TRƯỜNG ĐẠI HỌC FPT

# Health Care Personal

## Final Report

| HCP Team | | |
|---|---|---|
| Group members | Dang Tuan Anh | SE02643 |
| | Pham Thien Huong | SE02880 |
| | Duong Thai Duong | SE02570 |
| | Nguyen Cam Linh | SE02565 |
| | Le Khanh Thien | SE02447 |
| Supervisor | Pham Ngoc Ha | |
| Project code | HCP | |

**Monday, April 13, 2015**

# Acknowledge

HCP capstone project has consumed huge amount of work, research and dedication. But still, the implementation would not have been possible if we did not have a support of many individuals. Without them, this project would have been done with much more obstacles than it was .Therefore we would like to extend our sincere gratitude to all of them..

First of all, we would like to express our profound gratitude and deep regards to our lecturer Pham Ngoc Ha for his exemplary guidance, monitoring and constant encouragement throughout the Capstone project. The blessing, help and guidance given by him time to time shall carry us a long way in this project which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to Doctor Nguyen Van Doan, Military Medical Academy and Tran Huu Dat, Student of Ha Noi Medical University for their cordial support, valuable information and guidance about medical, which helped us in completing the foundation of information for this project.

We wish to express my sicere thanks to Mr Luong Trung Kien, Lecturer of FPT University for helping us in studying the technical which was used to buil this capstone project. And many thank to Tran Thanh Tung, Student of FPT University for supporting necessary technical informations.

We are also grateful to Nguyen Thuy Hanh and Pham Thi Minh Ngoc. Student of FPT University. We are extremly thankful and indebted to them for sharing expertise, and sincere and valuable guidance and encouragement extended to us.

We place on record, our sense of gratitude to one and all, who directly or indirectly, have lent us a hand to complete this project.

Lastly, we thank almighty, our family and friends for their constant encouragement without which this project would not be possible.

## TABLE OF CONTENTS

TABLE OF CONTENTS ..... 2
TABLE OF FIGURES ..... **Error! Bookmark not defined.**
DEFINITION AND ACRONYMS ..... 6
1 Introduction ..... 7
1.1 Project information ..... 7
1.2 Project members ..... 7
1.3 Purposes ..... 7
1.4 The existing system ..... 8
1.4.1 Medicine time ..... 8
1.5 Our proposal ..... 9
1.6 Product ..... 9
1.6.1 Judgment about personal health ..... 9
1.6.2 Remind prescription ..... 9
1.6.3 Finding near hospital in emergency ..... 9
2 Software Project Management ..... 10
2.1 Software Process Model ..... 10
2.2 Project Organization ..... 11
2.2.1 Organization Structure ..... 11
2.2.2 Project team ..... 11
2.2.3 External Interfaces ..... 13
2.3 Project Schedule ..... 13
2.3.1 Milestone and Deliverables ..... 13
2.3.2 Project schedule ..... 14
2.4 Resource ..... 15
2.4.1 Human resource ..... 15
2.4.2 Non-human resource ..... 15
2.5 Risk Management ..... 16
2.5.1 Risk register ..... 16

2.5.2 Probability/Impact matrix ..... 18
2.6 Communication Management ..... 18
2.6.1 Communication between Team Members ..... 18
2.6.2 Communication with the Supervisor ..... 18
2.6.3 Communication between Technical Supporter ..... 18
2.7 Configuration Management ..... 19
2.7.1 CI Identification & Naming convention ..... 19
2.7.2 Infrastructure ..... 20
2.7.3 Version Numbering Rule ..... 21
2.7.4 Directory structure ..... 22
2.7.5 Coding conventions ..... 23
2.7.6 Other CM Rules ..... 23
3 Software Requirement Specification ..... 24
3.1 User Requirements Specification ..... 24
3.1.1 Overall Description ..... 24
3.2 System Requirements Specification ..... 25
3.2.1 Interface Requirements ..... 25
3.2.2 Functional Requirements ..... 26
3.3 Non-Functional Requirements ..... 54
3.3.1 Usability ..... 54
3.3.2 Reliability ..... 54
3.3.3 Performance ..... 54
4 Software Design Specification ..... 55
4.1 Architecture Representation ..... 55
4.1.1 Overall System Architectural ..... 55
4.1.2 Application Architecture ..... 56
4.1.3 Architectural Goals & Constraints ..... 56
4.2 Architecture Design ..... 57
4.2.1 Logical view ..... 57
4.2.2 Process view ..... 105

4.3 Screen Design ..... 112
4.3.1 Screen flow ..... 112
4.3.2 Screen prototype ..... 114
4.4 Data design ..... 150
4.4.1 Entity Relationship Diagram ..... 150
4.4.2 Table ..... 151
5 Software source code ..... 158
5.1 Application ..... 158
5.2 Database ..... 159
6 Software Testing ..... 160
6.1 Test Strategy ..... 160
6.2 Test plan ..... 161
6.2.1 Test type ..... 161
6.2.2 Requirement for test ..... 162
6.2.3 Test stage ..... 164
6.2.4 Create test cases plan ..... 165
6.2.5 Execute test plan ..... 165
6.3 Testing Environments ..... 166
6.3.1 Software ..... 166
6.3.2 Hardware ..... 166
6.4 Test Milestones ..... 167
6.5 Test case ..... 168
6.5.1 Unit test case ..... 168
6.5.2 Test use case ..... 168
6.5.3 GUI test case ..... 169
6.5.4 Performance test case ..... 170
6.6 Test report ..... 170
6.7 Test Milestones ..... 172
6.8 Defect log ..... 172
7 Software User Manual ..... 173

7.1 Installation Guide ..... 173
7.1.1 English version ..... 173
7.1.2 Japanese version ..... 174
7.2 User Guide ..... 175
7.2.1 First time login ..... 175
7.2.2 Add new Patient ..... 176
7.2.3 Main screen ..... 177
7.2.4 Navigation drawer ..... 178
7.2.5 Patient ..... 179
7.2.6 Health judgement ..... 182
7.2.7 Recently status ..... 183
7.2.8 Reminder ..... 185
7.2.9 Hospital ..... 189
7.2.10 Help ..... 190
8 Appendixes ..... 191
8.1 Meeting minutes ..... 191
8.2 Progress report ..... 192
8.3 Q&A ..... 194
9 References ..... 195

# DEFINITION AND ACRONYMS

| Acronym | Definition | Note |
|---|---|---|
| **HCP** | Health Care Personal | |
| **PM** | Project Manager | |
| **QA** | Quality Assurance | |
| **PIC** | Person in charge | |
| **SRS** | Software Requirement Specification | |
| **AD** | Software Architecture Design | |
| **DD** | Software Detail Design | |
| **SDD** | Software Design Description | |
| **DB** | Database | |
| **UT** | Unit Test | |
| **IT** | Integration Test | |
| **TC** | Test case | |
| **JP** | Japanese | |

# 1 Introduction

## 1.1 Project information

- **Project Name:** Health Care Personal
- **Project Code:** HCP
- **Project Type:** Android application
- **Project Category:** New development
- **Timeline:** from 5th January 2015 to 3rd May 2015

## 1.2 Project members

➢ **Supervisors**

| Full name | Phone | E-Mail | Title |
|---|---|---|---|
| **Pham Ngoc Ha** | 0988623000 | hapn@fsoft.com.vn | Lecturer |

➢ **Team members**

| Full name | Roll no. | Phone | E-mail | Position |
|---|---|---|---|---|
| **Đặng Tuấn Anh** | SE02643 | 01694845428 | Anhdtse02643@fpt.edu.vn | Leader |
| **Lê Khánh Thiện** | SE02447 | 01679957600 | Thienlkse02447@fpt.edu.vn | Member |
| **Đường Thái Dương** | SE02570 | 01245888062 | Duongdtse02570@fpt.edu.vn | Member |
| **Nguyễn Cẩm Linh** | SE02565 | 01662226798 | Linhncse02565@fpt.edu.vn | Member |
| **Phạm Thiên Hương** | SE02880 | 0974027164 | Huongptse02880@fpt.edu.vn | Member |

## 1.3 Purposes

Nowaday, Smartphone has been developed explosively. There are a lot of applications on Smartphone which made our life become better and better. The application about taking care of personal health became more and more needful and indispensable.

The life's standard got higher, and healthy become the issue which people care about the most. In this model life, people don't have much free time , moreover , going to hospital in Viet Nam takes too much time, and be so complicate. That make people become more careless about their heathy and lazy to keeping track of their own status.

For that purpose, we developed this application to help solving following problems:

- The application let user input their health's information and base on that, HCP application will give you the judgement about user's status.
- The application also help people tracking the prescription.
- You can use our application to find the wat to the closest hospital. It will be helpfull is emergency situation.

## 1.4 The existing system

### 1.4.1 Medicine time

Link: https://play.google.com/store/apps/details?id=com.jmsoft.medicinereminder

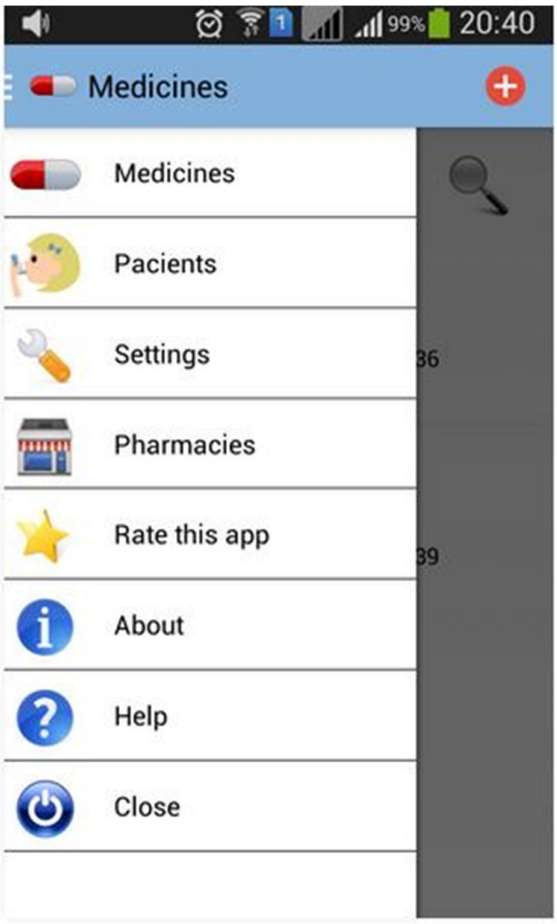



*Figure 1-1 Medicine time 1*

*Figure 1-2 Medicine time 2*

**Pros:**

- The application support user with their prescription
- Application's interface are friendly to users.

**Cons:**

- Some another necessary functionalities such as give judgement about health's status, finding hospital in normal or emergency situation,... are still missing.

## 1.5 Our proposal

The purpose of this application is to help user have methods to prevent early pathological signs such as diabetes, high or low blood pressure and have healthy life. In the other hands, this application also remind user about doctor's prescription to drink medicine on time, name and numberof tablet patient have to drink each time,and notify when the pills are running out.

## 1.6 Product

Our product have 3 main functions include:

### 1.6.1 Judgment about personal health

- At first, user to enter the personal health indicators such as blood pressure, heart beat and glycemic index. The information for each field is fixed in a dropdown list so that user don't need to input anything, just need to choose the information which suit their index.
- Applications will give overall assessment of the health status of the user based on the index chosen and the information of patient to have certain information about personal health.
- This function also auto save a record of the chosen index and the advice was given into databse, so that user can track their own condition. The history of this function will be display as a list of date and time, and when user click in a row, it will display all contain informations.

### 1.6.2 Remind prescription

- The user will enter the information of their prescription from the doctor. Information includes medicine's name, drinking time, quantity, number of tablet need to take each time, daily interval and notes if it has.
- The information will be saved to the database. User can see list of reminder in two options: All reminder or Today's Reminder. All Reminder will display all the reminder, Today's Reminder will display reminders which will be alarmed today. The reminder be displayed is depend on the current selected Patient.
- The application will update the amount of medicine remaining after each prompt, and prompt the user to the information has been entered.

### 1.6.3 Finding near hospital in emergency

- This function supports users to search and browse quickly to the hospital near the current location of the user in case of emergency.
- On the display information screen of the hospital will have the function to call the hospital.

# 2 Software Project Management

## 2.1 Software Process Model

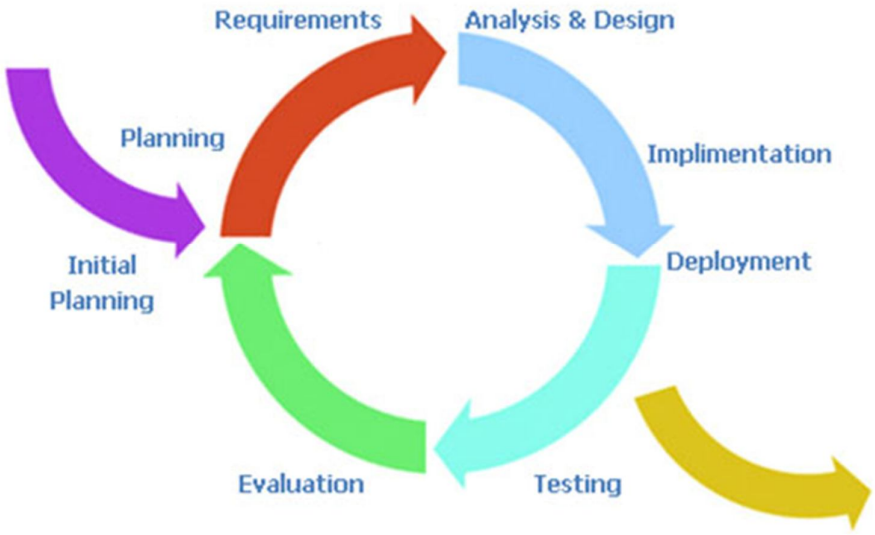



*Figure 2-1 Project process model*

We choose Iterative model for development process of this project. Iterative is a life cycle model does not attempt to start with a full specification of requirements. The development starts by specify and implement a part of the software, then it will be reviewed to take advantages and disadvantages of the product. This process is then repeated to produce a new version of the software at the end of each iteration of the model.

This model have some advantages that we feel appropriate for our project:

- ✓ Some functions can be developed early & quickly.
- ✓ Easy & less costly to change requirements.
- ✓ Results are obtained early and periodically.
- ✓ Parallel development can be planned.
- ✓ Testing and debugging during smaller iteration is easy.
- ✓ During life cycle software is produced early which facilitates user evaluation and feedback.

## 2.2 Project Organization

### 2.2.1 Organization Structure

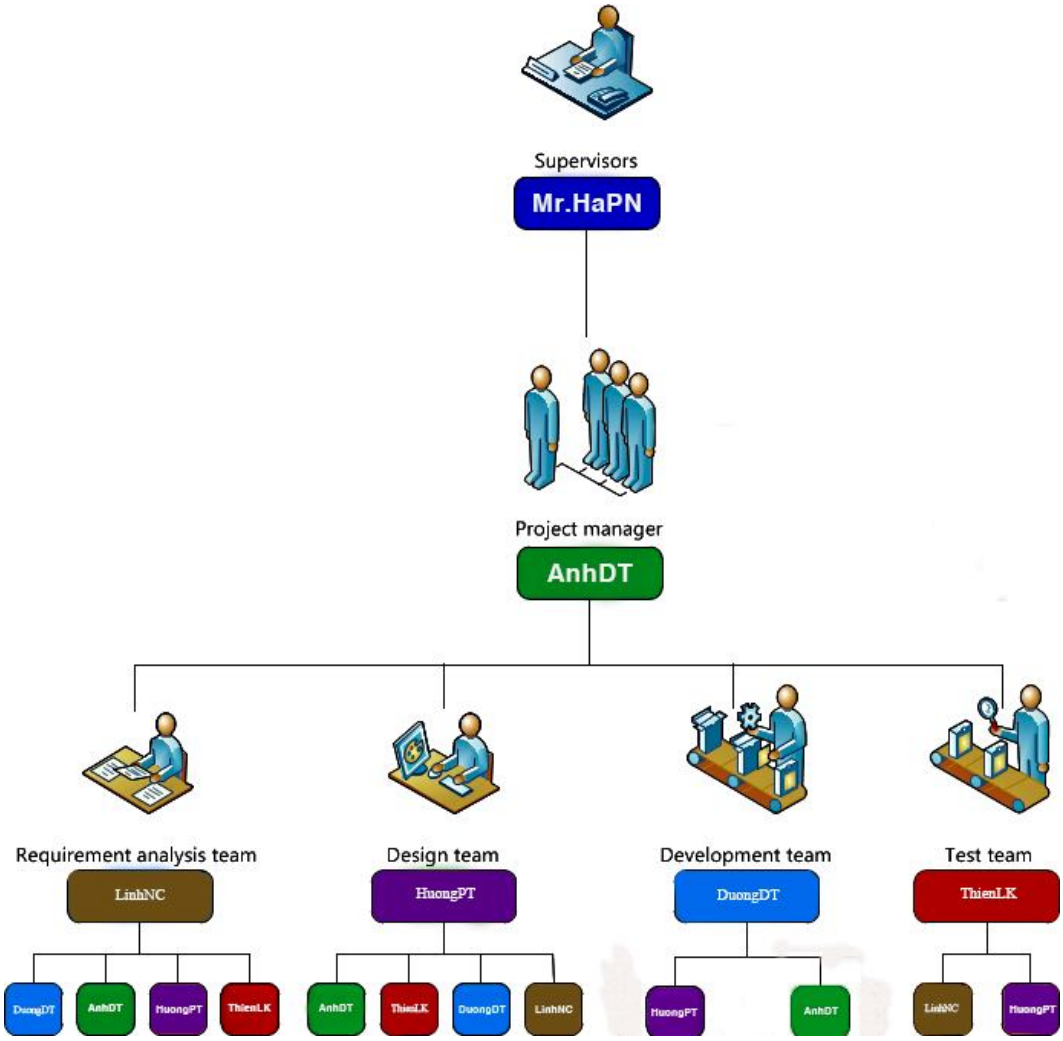



*Figure 2-2 Project organizational structure*

### 2.2.2 Project team

**TABLE 2-1 Roles and Responsibilities**

| Full name | Roles in project | Responsibilities |
|---|---|---|
| **Dang Tuan Anh** | Project manager | Develop project plan<br>Manage project stakeholders, project team and resolve conflicts<br>Manage project schedule, project risk<br>Communication with members and supervisor, include weekly plan, weekly status report, hold meetings and assure to submit reports to supervisor on schedule.<br>Manage data synchronization in project |
| | Requirement analysis | Create and co-operate with members to develop SRS and SDD |
| | Detail designer | |
| | Developer | Co-operate with members to coding |
| **Nguyen Cam Linh** | Requirement analysis leader | Develop SRS document<br>Co-operate with members to develop SRS |
| | Detail designer | Co-operate with members to develop SDD |
| | Tester | Create test cases for the system<br>Execute test to ensure all functions fulfill requirements |
| **Pham Thien Huong** | Requirement analysis | Co-operate with members to develop SRS |
| | Design leader | Conduct overall business process<br>Co-operate with members to develop SDD |
| | Database designer | Perform database analysis and design<br>Create database base on Design document |
| | Developer | Co-operate with members to coding |
| **Duong Thai Duong** | Requirement analysis | Co-operate with members to develop SRS |
| | Screen designer | Co-operate with members to develop GUI |
| | Technical leader | Provide suitable technology solutions<br>Conduct research and demo function related to the project<br>Configure coding environment |
| **Le Khanh Thien** | Test leader | Develop the test plan based on project plan, SRS and Design documents<br>Delegate testing tasks to testers and manage the rate of progress<br>Create test data for all functions<br>Write test report |
| | Requirement analysis | Co-operate with members to develop SRS |

| | | |
|---|---|---|
| | Detail designer | Create document format standards and coding conventions.<br>Co-operate with members to develop DD |
| | Tester | Execute test to ensure all functions fulfill requirements |

### 2.2.3 External Interfaces

#### 2.2.3.1 FPT University's Interfaces:

| Department | Contact Person (name-position) | Contact address (email, telephone) | Responsibility |
|---|---|---|---|
| **Management and Training Department** | Nguyen Thi Mai Phuong | hantm@fpt.edu.vn | Receive capstone project register |
| **Management and Training Department** | Luong Trung Kien | kienlt@fpt.edu.vn | Support Android Technical |
| **FU Student** | Tran Thanh Tung | 01672852936 | Support Android Technical |

#### 2.2.3.2 Others

| Department | Contact Person (name-position) | Contact address (email, telephone) | Responsibility |
|---|---|---|---|
| **Hanoi Medical University** | Tran Huu Dat | N/A | Support medical knowledge |
| **Viet Nam Military Medical University** | Dr. Nguyen Van Doan | N/A | Support medical knowledge |

## 2.3 Project Schedule

### 2.3.1 Milestone and Deliverables

**TABLE 2-2 Milestone and Deliverables**

| No. | Phase | Deliverable/ Milestone | Delivery Date | Delivery Location |
|---|---|---|---|---|
| **1** | Initiation | Project identification | 2015/01/05 | Project team |
| **2** | Definition | Project Plan | 2015/01/13 | Supervisor |
| **3** | | Requirement Specification | 2015/01/24 | Supervisor |
| **4** | Solution | UI Design | 2015/01/29 | Project team |

| | | | | |
|---|---|---|---|---|
| **5** | | Database Design | 2015/01/31 | Project team |
| **6** | | Architectural Design | 2015/02/09 | Supervisor |
| **7** | Construction | Test case | 2015/03/05 | Supervisor |
| **8** | | Complete code | 2015/03/18 | Supervisor |
| **9** | | Complete system test | 2015/04/15 | Supervisor |
| **10** | | Complete full source code | 2015/04/15 | Supervisor |
| **11** | Termination | Complete fully project product | 2015/04/18 | Supervisor |
| **12** | | Complete Presentation slide | 2015/04/20 | Supervisor |
| **13** | | Deliver Final Report & CD | 2015/04/24 | Supervisor |
| **14** | | Project completed | 2015/04/27 | Supervisor |
| **15** | | Defend project | 2015/05/06 | |

### 2.3.2 Project schedule

| | Task Mode | Task Name | Duration | Start | Finish | Predecessors | Resource Names | Add New Column |
|---|---|---|---|---|---|---|---|---|
| 1 | | Initiation | 1 day | Mon 1/5/15 | Mon 1/5/15 | | | |
| 2 | | Kick - off Meeting | 0.25 days | Mon 1/5/15 | Mon 1/5/15 | | | |
| 3 | | Register Capstone Project | 0.25 days | Mon 1/5/15 | Mon 1/5/15 | 2 | Dang Tuan Anh | |
| 4 | | Develop project charter | 0.25 days | Mon 1/5/15 | Mon 1/5/15 | 3 | Dang Tuan Anh | |
| 5 | | Project charter approved | 0.25 days | Mon 1/5/15 | Mon 1/5/15 | 4 | Supervisor Pham N | |
| 6 | | Definition | 16.5 days | Tue 1/6/15 | Sat 1/24/15 | | | |
| 7 | | **Develop project plan** | **6 days** | **Tue 1/6/15** | **Tue 1/13/15** | | | |
| 8 | | Create task list | 0.5 days | Tue 1/6/15 | Tue 1/6/15 | | | |
| 9 | | Create estimates | 0.5 days | Tue 1/6/15 | Tue 1/6/15 | 8 | | |
| 10 | | Create network diagram (PERT) | 0.25 days | Wed 1/7/15 | Wed 1/7/15 | 9 | | |
| 11 | | Create work breakdown structure (WBS) | 0.5 days | Wed 1/7/15 | Wed 1/7/15 | 10 | Dang Tuan Anh | |
| 12 | | Create schedule (GANTT) | 1.5 days | Wed 1/7/15 | Fri 1/9/15 | 11 | Dang Tuan Anh | |
| 13 | | Create milestone plan | 0.5 days | Fri 1/9/15 | Fri 1/9/15 | 12 | Dang Tuan Anh | |
| 14 | | Create organization plan | 0.25 days | Fri 1/9/15 | Fri 1/9/15 | 13 | Dang Tuan Anh | |
| 15 | | Create resource assignments | 0.5 days | Sat 1/10/15 | Sat 1/10/15 | 14 | Dang Tuan Anh | |
| 16 | | Develop communications plan | 0.5 days | Sat 1/10/15 | Sat 1/10/15 | 15 | Dang Tuan Anh | |
| 17 | | Develop quality plan | 0.5 days | Mon 1/12/15 | Mon 1/12/15 | 16 | Dang Tuan Anh | |
| 18 | | Develop risk management plan | 0.5 days | Mon 1/12/15 | Mon 1/12/15 | 17 | Dang Tuan Anh | |
| 19 | | Project plan approved | 0 days | Tue 1/13/15 | Tue 1/13/15 | 18 | | |
| 20 | | **Develop requirement document** | **12 days** | **Sat 1/10/15** | **Sat 1/24/15** | | | |
| 21 | | Prepare research instruments | 1 day | Sat 1/10/15 | Mon 1/12/15 | | | |
| 22 | | **Create requirements document** | **8.5 days** | **Mon 1/12/15** | **Wed 1/21/15** | | | |
| 23 | | Create user requirements | 2 days | Mon 1/12/15 | Wed 1/14/15 | | Nguyen Cam Linh | |
| 24 | | Create system requirements | 3 days | Wed 1/14/15 | Sat 1/17/15 | 23 | Nguyen Cam Linh | |
| 25 | | Create screen prototype | 1 day | Sat 1/17/15 | Mon 1/19/15 | 24 | Duong Thai Duong | |
| 26 | | Create database requirements | 2 days | Mon 1/19/15 | Wed 1/21/15 | 25 | Le Khanh Thien | |
| 27 | | Create non-functional requirements | 0.5 days | Wed 1/21/15 | Wed 1/21/15 | 26 | Le Khanh Thien | |

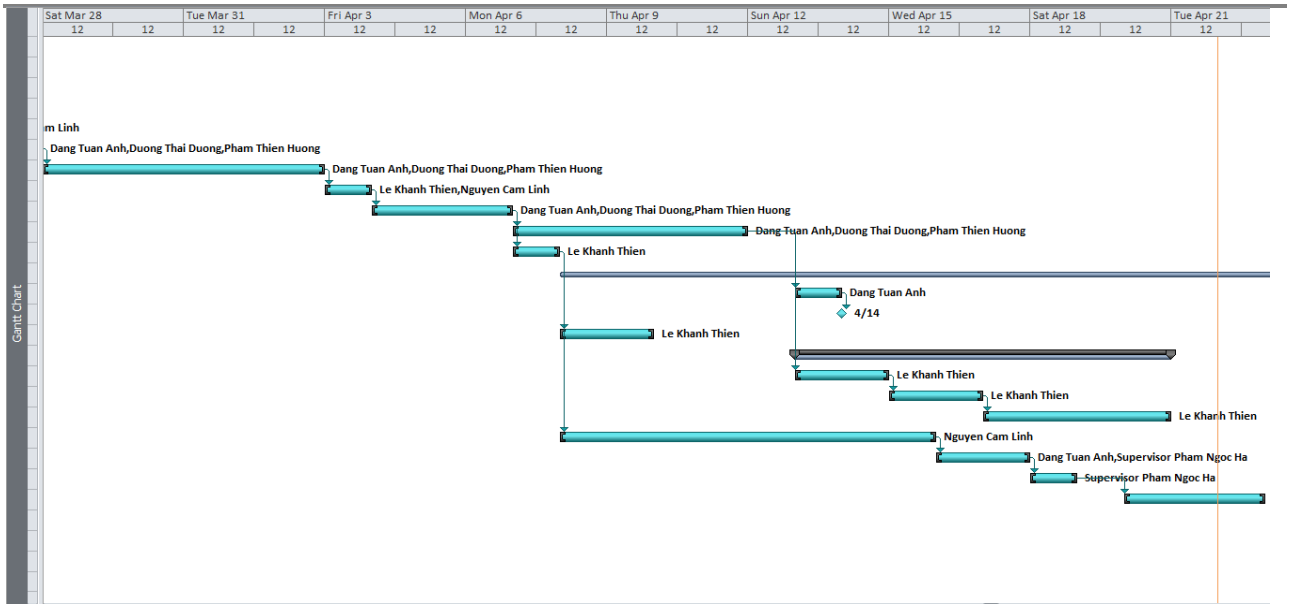



*Figure 2-3 Project organizational structure*

## 2.4 Resource

### 2.4.1 Human resource

- Team members
- Supervisor
- Android supporter
- Medical supporter

### 2.4.2 Non-human resource

- Equipment: Desktop, Laptop, Android smart phone
- Building: Library, Hoa Lac campus

## 2.5 Risk Management

### 2.5.1 Risk register

**Table 2-3 Risk register**

| No | Risk | Description | Category | Avoidance | Contingency | Status |
|---|---|---|---|---|---|---|
| **1** | Key member leaving | Key member leaves | People risk | Have a suitable treatment policy | Find other person | Open |
| **2** | New technical | New technical is applied | Technology risk | Training | Ask for expert | Open |
| **3** | Change requirement | Requirement is changed when designing/ coding or testing | Structure risk | Discuss and define clearly requirement from start | Change all related resource | Open |
| **4** | Not suitable schedule | Schedule is not suitable, duration is too long or too short | Structure risk | Use tool when create schedule, ask PIC about duration to complete each task | Change schedule, add more effort | Open |
| **5** | Miss requirement | The product of project cannot meet fully requirements | Structure risk | Analyze requirement clearly right from the start, review and confirm SRS | Add more into design, code, test case | Open |
| **6** | Lost data | Data is lost by laptop break down or OS is error | Technology risk | Back up data/ sent to other server/ computer | Use tool to restore data | Open |
| **7** | Lazy member | Member doesn't complete own works, irresponsible | People risk | Create rule, deadline | Punishment | Open |
| **8** | Late deadline | Work/ process is late in comparison with deadline | Structure risk | Split the deadline, divide clearly work and responsibility | Over time, add more effort | Open |

| | | | | | | |
|---|---|---|---|---|---|---|
| **9** | Lack of experience | Member lack of experience in working with development project process | People risk | Training | Other member support | Open |
| **10** | Lack skills | Team members lack skills about technical and soft skills | Technology risk | Training | Ask for expert, Other member support | Open |
| **11** | Technical issues | Rising technical issues lead project delay | Technology risk | Training | Ask for expert, overtime | Open |
| **12** | Big scope | Scope is bigger than ability that team can do | Structure risk | Discuss scope careful right from the start | Add more people/ cut down requirement | Open |
| **13** | Bad leader | Team leader is conservative, or the ability is bad, or irresponsible… | People risk | Choose the person who has ability, experience, and has a sense of responsibility | Change leader | Open |
| **14** | Bad infrastructure | Operating system is error, or laptop break down | Technology risk | Prepare carefully | Buy/ rent/ borrow | Open |
| **15** | Bad relationship | Team members may be conflict with each other that cause project delay | People risk | Usually team building to improve solidarity | Reconcile | Open |
| **16** | Not good healthy | Member is sick when working | People risk | Stay healthy, limit over time | Change response | Open |

### 2.5.2 Probability / Impact matrix

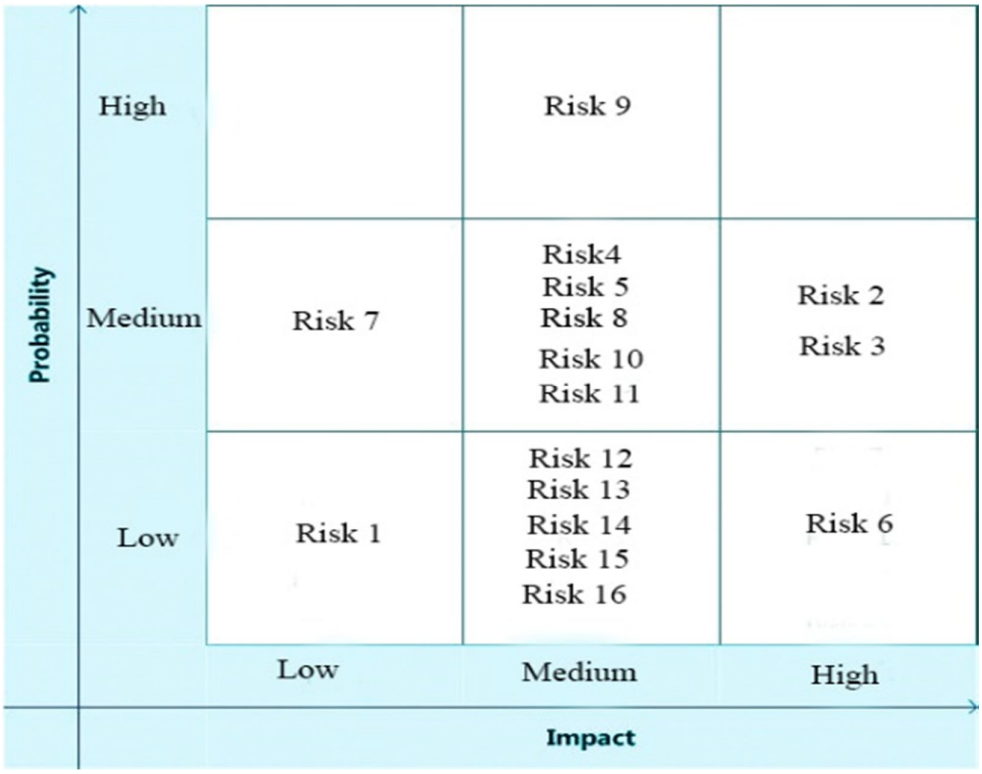



*Figure 2-4 Risk Probability/Impact matrix*

## 2.6 Communication Management

### 2.6.1 Communication between Team Members

- **Face-to-face meeting**: five times a week, from Monday to Friday at 7:00 PM. This is the most effective way to communicate, and we can solve problems easily.
- **Email and message**: Gmail, Skype and Facebook are used to communicate online. It helps us to keep track of team's progress.
- **Mobile phone**: In an emergency situation, we use mobile phone to contact directly with other members.
- **Collaboration tool**: Google drive is used for managing document and source code.

### 2.6.2 Communication with the Supervisor

We have a weekly meeting on every Tuesday, so the supervisor can keep track of the team's progress.

### 2.6.3 Communication between Technical Supporter

- **Face-to-face meeting:** We have a face-to-face meeting twice at the beginning of the project to helps members to training the new technical. We also have to arrange an appointment for some technical problem cases happen.
- **Mobile phone:** is used to arrange appointment and also contact directly to Supporters to solve problems.

## 2.7 Configuration Management

### 2.7.1 CI Identification & Naming convention

**Table 2-4 CI Identification & Naming convention**

| No | Configuration Items | Naming convention |
|---|---|---|
| **Project Management** | | |
| **1** | Project Plan | HCP_Project Plan_v.[version number]<br>For example: HCP_Project Plan_v1.0 |
| **Requirement & Design** | | |
| **2** | Software Requirements Specification | HCP_Software Requirements Specification_v.[version number]<br>For example: HCP_Software Requirements Specification_v1.0 |
| **3** | Software Design Description | HCP_Software Design Description_v.[version number]<br>For example: HCP_ Software Design Description_v1.0 |
| **4** | Screen Design | HCP_Screen Design_v.[version number]<br>For example: HCP_Screen Design_v1.0 |
| **5** | Data Design | HCP_ Data Design_v.[version number]<br>For example: HCP_ Data Design_v1.0 |
| **Source Code** | | |
| **6** | Source Code | HCP_Source Code_ v.[version number] [Tested/Untested] For example: HCP_Source Code_v1.0 Tested |
| **Support Document** | | |
| **7** | User Manual | HCP_User Manual_v.[version number]<br>For example: HCP_User Manual_v.[version number] |
| **Test** | | |
| **8** | Test Plan | HCP_Test Plan_v.[version number]<br>For example: HCP_Test Plan_v1.0 |
| **9** | Test Case | HCP_Test Case_v.[version number]<br>For example: HCP_Test Case_v1.0 |
| **10** | Test Result | HCP_Test Report_v.[version number]<br>For example: HCP_Test Report_v1.0 |
| | | |

| | | |
|---|---|---|
| **Process** | | |
| **11** | Guideline | HCP_ [Name Of Guideline] Guideline_v.[version number]<br>For example: HCP_Unit Test Guideline_v1.0 |
| **12** | Convention | HCP_[Name Of Convention] Conventions_v.[version number]<br>For example: HCP_Coding Conventions_v1.0 |
| **13** | Checklist | HCP_[Name Of Checklist] Checklist_v.[version number]<br>For example: HCP_Review Code Checklist_v1.0 |
| **File Type** | | |
| **14** | MS Word | *.doc, *.docx |
| **15** | MS Excel | *.xls, *.xlsx |
| **16** | MS PowerPoint | *.ppt, *.pptx |
| **17** | MS Project | *.mpp |
| **18** | Enterprise Architect | *.eap |
| **19** | Images | *.png or *.jpg or *.jpeg or *.bmp or *.gif |

### 2.7.2 Infrastructure

Below is the list of tools and infrastructure requirements needed for development environment.

#### *2.7.2.1 Software*

**Table 2-5 Software needed for project**

| Category | Software name | Version |
|---|---|---|
| **Operating System** | Microsoft Windows 8 | Professional |
| **Office tools** | Microsoft Office | 2010, 2013 |
| **Task tracking** | Microsoft Project | 2010 |
| **Design tools** | Justinmind prototyper | |
| | www.draw.io | |
| | Adobe Photoshop | CS6 |
| | Enterprise Architecture | 9.0 |
| **Development framework** | Microsoft .NET | 4.0 |
| | JDK | 7 |

| | | |
|---|---|---|
| **Development tools** | Eclipse | Newest |
| | Notepad++ | Newest |
| **Database tools** | SQLite | |
| **Source version control** | Google drive | |
| **Documentation** | Microsoft Office Word | 2010, 2013 |
| | Microsoft Office Excel | 2010, 2013 |
| | Microsoft PowerPoint | 2010, 2013 |
| **Support tool** | Skype | |

#### 2.7.2.2 Hardware

- Personal computer for developing/testing with the minimum configuration: 2GB Ram, 80GB of hard disk, Processor Intel Core i3
- Internet network connection with minimum speech 512kbit/s.

#### 2.7.2.3 Others infrastructures

- A room for team's meeting.
- Internet and mobile phone services are needed for communication.
- Android smartphone for test.

### 2.7.3 Version Numbering Rule

➢ **For documents:**

Each file has a version number as part of its identity. This version number is physically represented as a 2-part string with the following format:

**<version>.<revision>**

For example, version 1.0 indicates 1 as the version, and 0 as the revision number.

The original version will be numbered 0.1. Subsequent revisions will be numbered 0.2, 0.3 and so on. The approved version will be 1.0.

- **Version number:** appears to the left of the decimal. It is changed only when the core content of the item is significance changed. For example: when an item is completely overhauled, with substantial internal changes, the version 1.0 would become version 2.0.
- **Revision number:** appears to the right of the decimal. It is changed when the existing content is changed, but the main (or core) content is remained. The normal sequence of revision is 1.1, 1.2, and so on.

➢ **For Software source files:**

Software executable and support files are generally identified by name and version number. The version number is physically represented as a 3-part string with the following format:

**<version>.<revision><update>**

For example, version 1.1a indicates 1 as the version, 0 as the revision number, and a as the update level.

- **Version number:** appears to the left of the decimal. It is changed only when the core content of the item is significance changed, as when moving from one are of the development tool to another, when an application is completely overhauled, or the user interface changes fundamentally. In this case, version 1.1a would become version 2.0.
- **Revision number:** appears to the right of the decimal. It is changed when new features, functionality or other content are added or significantly changed. In normal case, the core architecture or user interface have been extended or limited in some manner. The most common reason for changing the revision number is adding a new module or other functionality to the software. The normal sequence of revision is 1.0, 1.1 and 1.2 and so on.
- **Update level:** is appended or incremented when the only change to the software item is to correct one or more defects, without the addition of any new function. Version 1.1 would become v1.1a, 1.1b and so on. This updating is overridden when a combination revision, involving bug fixes and new feature additions, is performed. In such a case, the software revision number is incremented and any update indicator is dropped, as in v1.1b to 1.2

### 2.7.4 Directory structure

**Table 2-6 Directory structure**

| Main folder | Sub-folder | Purpose |
|---|---|---|
| **Archive** | Baseline Name | To released versions of CIs at baselines |
| **Audit** | | Store check list file |
| **Image** | Image category name<br>Example: Prototype | Store image of project |
| **Documents** | Meeting minutes | Store project meeting minutes |
| | Q&A | Store Q&A Management Sheet version English and version Japanese |
| | Weekly Report | Store project weekly report |
| | Defect log | Store bug list & comment list file |

| | | |
|---|---|---|
| **Deliverable** | Report 1 | Store final deliverables of phase 1 |
| | Report 2 | Store final deliverables of phase 2 |
| | Report 3 | Store final deliverables of phase 3 |
| | Report 4 | Store final deliverables of phase 4 |
| | Report 5 | Store final deliverables of phase 5 |
| | Report 6 | Store final deliverables of phase 6 |
| **Plan** | | Store project plan, Task list… |
| **Resource** | Reference materials | Store Reference materials needed in project |
| | Template | Store template needed in project |
| | Tool | Store tool needed in project |
| **WIP** | mmddyy<br>( m=month, d=day, y= year )<br>(Example: 210115) | Tracking working process<br>Store project deliverable |
| **Working space** | Each team members has a folder.<br>( Example: AnhDT) | Team member's working area |

### 2.7.5 Coding conventions

Coding conventions are important to programmers for a number of reasons:

- ✓ They create a consistent look to the code, so that readers can focus on content, not layout.
- ✓ They enable readers to understand the code more quickly by making assumptions based on previous experience.
- ✓ They facilitate copying, changing, and maintaining the code.

This project use Android coding conventions following by Google.

Refer Project Conventions Coding Standards Java/Android to see detail.

### 2.7.6 Other CM Rules

**Email subject naming convention**: All email related to the HCP project must have prefix [HCP].

For example: [HCP] Progress report 24/01/2015

**Document changing rule:** When a member want to modify a document, he/she must update version of that document with appropriate description for the modification.

# 3 Software Requirement Specification

## 3.1 User Requirements Specification

### 3.1.1 Overall Description

HCP android application supports user knows basic information about their own health through the input parameters from the screen and provides the function prompts the user to use prescription drugs by doctors and management of medicine quantity.

HCP applications also support function displays hospitals near user current position with information and directions to get there.

**TABLE 3-1 Application features**

| Function group | Function |
|---|---|
| **First time login** | |
| **Patient Management** | View patient list |
| | Add new patient |
| | Switch patient |
| | Patient detail information |
| | Edit patient information |
| | Delete patient |
| **Health judgment** | Health judgment |
| **Recently status** | Recently status detail |
| **Reminder** | View reminders list |
| | Add new reminder |
| | Reminder detail information |
| | Edit reminder |
| | Delete reminder |
| **Hospital** | |
| **Help** | Help Detail |
| **About us** | |

## 3.2 System Requirements Specification

### 3.2.1 Interface Requirements

#### 3.2.1.1 User Interfaces

- User interface must be designed based on screen design document.
- User interface must attract the attention of users to the images, information display in the screen.
- Each UI element must be arranged logically, not overlapped over each other, allowing users access easily.
- The look and feel of the mobile app should be based on user requirements. Page layout and navigation should be as intuitive and easy as possible.

#### 3.2.1.2 Hardware Interfaces

- The device has an internet connection (3G, GPRS) and GPS.
- The minimum configuration for your Android device: 512MB RAM, storage of memory 4GB.
- Minimum OS: Android 4.0

#### 3.2.1.3 Software Interfaces

- Application running on the android operating system developed by tools Eclipse.
- Application use management system data base SQLite for android.

#### 3.2.1.4 Communications Interfaces

- Use network to receive map and locate location.

### 3.2.2 Functional Requirements

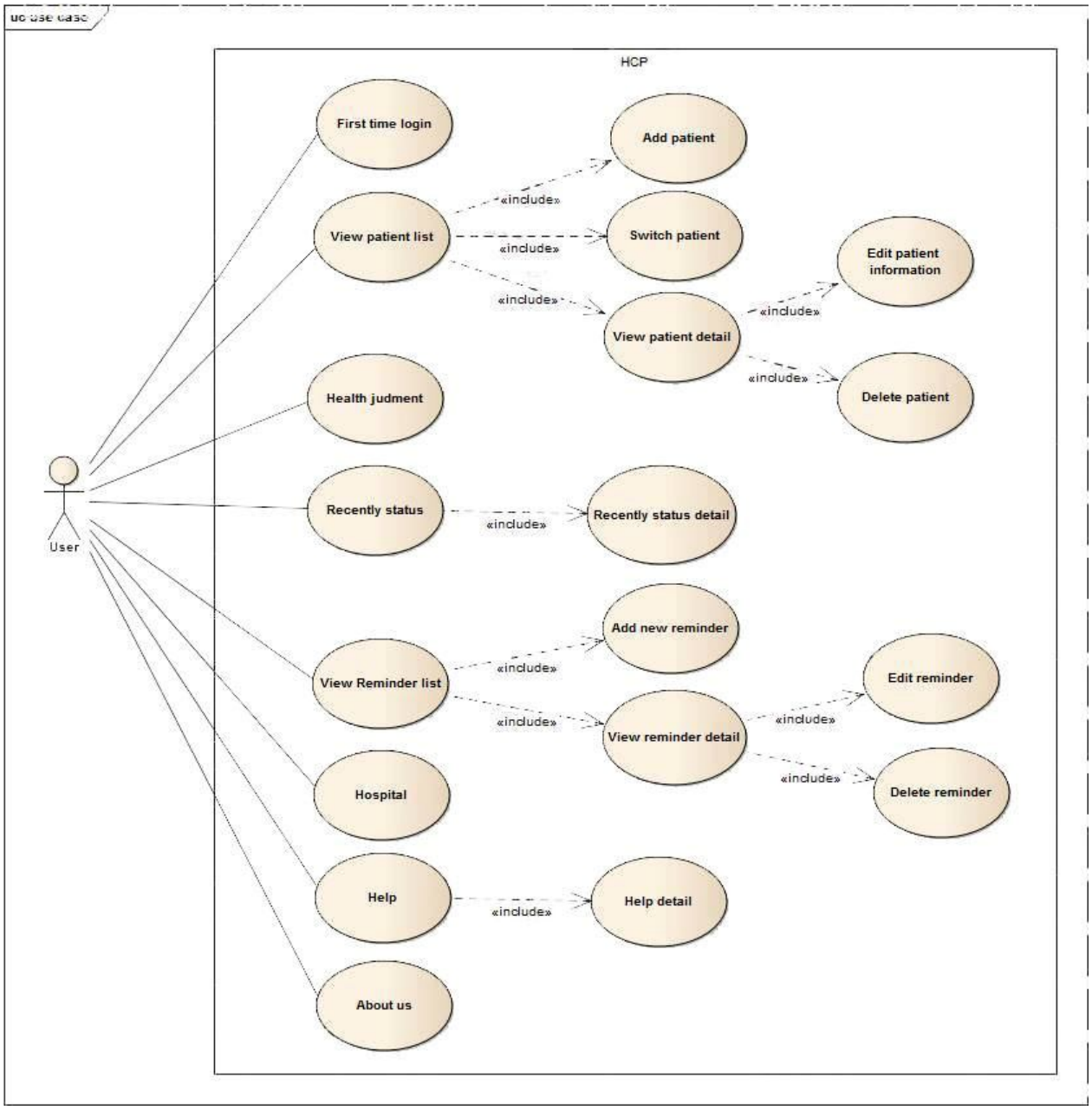



Figure 3-1 Use-case diagram

TABLE 3-2 Application Use-cases

| No. | Use-case name | Use-case no. |
|---|---|---|

| | | |
|---|---|---|
| **1** | First time login | UC-01 |
| **2** | View patient list | UC-02 |
| **3** | Add new patient | UC-03 |
| **4** | Switch patient | UC-04 |
| **5** | View patient detail | UC-05 |
| **6** | Edit patient information | UC-06 |
| **7** | Delete patient | UC-07 |
| **8** | Health judgment | UC-08 |
| **9** | Recently status | UC-09 |
| **10** | Recently status detail | UC-10 |
| **11** | View reminder list | UC-11 |
| **12** | Add new reminder | UC-12 |
| **13** | View reminder detail | UC-13 |
| **14** | Edit reminder | UC-14 |
| **15** | Delete reminder | UC-15 |
| **16** | Hospital | UC-16 |
| **17** | Help | UC-17 |
| **18** | Help detail | UC-18 |
| **19** | About us | UC-19 |

#### 3.2.2.1 First time login

TABLE 3-3 Use case First time login specification

| USE-CASE FIRST TIME LOGIN SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-1 | **Use-case version** | 0.1 |
| **Use-case name** | First time login | | |
| **Author** | LinhNC | | |
| **Date** | 28/01/2015 | **Priority** | High |
| **Actor:**<br>User | | | |
| **Summary:**<br>When user use the application for the first time “Thêm bệnh nhân” screen will display to register information of first patient. | | | |
| **Goal:**<br>Information of first patient is saved into database. | | | |
| **Triggers:**<br>User open HPC application at the first time. | | | |
| **Pre-conditions:**<br>N/A | | | |
| **Post-conditions:**<br>Main screen display. | | | |

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| **1** | User open HPC application. | Application display “Thêm bệnh nhân” screen |
| **2** | User enter information into fields at the screens. | |
| **3** | User touch “OK” button on “Thêm bệnh nhân” screen. | - Application check condition:<br>+ All field must be filled.<br>- If User input incorrect information application show error notification. (ER_001)<br>- If User input all correct information application save user’s information to database and return main screen. |

**Alternative Scenario:**

| Step | Action | Response |
|---|---|---|
| **2** | User touch “Nhập lại” button on “Thêm bệnh nhân” screen. | All information on fields at the screen be cleaned. |
| **2** | User touch “Back” button on device. | Application is closed. |

**Exceptions:**

| Step | Action | Response |
|---|---|---|
| **3** | Input information, touch on “OK” button | ER_001: Show message error below the field which have incorrect information. |

**Business Rules:**

N/A

### 3.2.2.2 Patient management

#### 3.2.2.2.1 Patient list

TABLE 3-4 Use case Patient list specification

| USE-CASE PATIENT LIST SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-2 | **Use-case version** | 0.1 |
| **Use-case name** | Patient list | | |
| **Author** | LinhNC | | |
| **Date** | 28/01/2015 | **Priority** | High |
| **Actor:** User | | | |
| **Summary:** In this use-case, User view list of all patients already have on application. | | | |
| **Goal:** Application show patient list. | | | |
| **Triggers:** User touch on name of patient which is displayed on tool bar of screen or touch on "Quản lý" on Navigation drawer. | | | |
| **Pre-conditions:** N/A | | | |
| **Post-conditions:** List user be displayed on screen. | | | |

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch on name of patient on tool bar of screen or touch on "Quản lý" on Navigation drawer. | Application show "Danh sách bệnh nhân" screen with list of all patients already have on application. |

| |
|---|
| **Alternative Scenario:**<br>N/A |
| **Exceptions:**<br>N/A |
| **Business Rules:**<br>N/A |

#### 3.2.2.2.2 Add new patient

**TABLE 3-5 Use case Add new patient specification**

| USE-CASE ADD NEW PATIENT SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-3 | **Use-case version** | 0.1 |
| **Use-case name** | Add new patient | | |
| **Author** | LinhNC | | |
| **Date** | 28/01/2015 | **Priority** | High |
| **Actor:**<br>User | | | |
| **Summary:**<br>User add new patient | | | |
| **Goal:**<br>Application add new patient with information to database. | | | |
| **Triggers:**<br>User touch on add button on “Danh sách bệnh nhân” screen. | | | |
| **Pre-conditions:**<br>Application is showing “Danh sách bệnh nhân” screen. | | | |
| **Post-conditions:**<br>Application show “Danh sách bệnh nhân” screen with new user. | | | |
| | | | |

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| **1** | User touch on button add on "Danh sách bệnh nhân" screen. | Application show "Thêm bệnh nhân" screen. |
| **2** | User enter information into fields at the screens. | |
| **3** | User touch on button OK on "Thêm bệnh nhân" screen. | - Application check condition:<br>+ All field must be filled.<br>+ Patient name have maximum length is 30 and must not have special digit or number (@!~#$%^&*?<>:;)<br>- If User input incorrect information application show error notification. (ER_001)<br>- If User input all correct information application save patient's information to database and return Main screen. |

**Alternative Scenario:**

| Step | Action | Response |
|---|---|---|
| **2, 3** | User touch "Nhập lại" button on "Thêm bệnh nhân" screen. | All information on fields at the screen be cleaned. |
| **2, 3** | User touch "Back" button on device. | Application show "Danh sách bệnh nhân" screen. |

**Exceptions:**

| Step | Action | Response |
|---|---|---|
| **3** | Input information, touch on OK button | ER_001: Show message error below the field which have incorrect information. |

**Business Rules:**

N/A

#### 3.2.2.2.3 Switch patient

**TABLE 3-6 Use case Switch patient specification**

| USE-CASE SWITCH PATIENT SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-4 | **Use-case version** | 0.1 |
| **Use-case name** | Switch patient | | |
| **Author** | LinhNC | | |
| **Date** | 28/01/2015 | **Priority** | High |
| **Actor:** User | | | |
| **Summary:** In this use-case, User change current patient to other patient. | | | |
| **Goal:** Patient is changed. | | | |
| **Triggers:** User touch on name of patient at tool bar or “Quản lý” item on Navigation drawer. | | | |
| **Pre-conditions:** N/A | | | |
| **Post-conditions:** Application display Main screen with other patient. | | | |

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch on name of patient at tool bar or “Quản lý” item on Navigation drawer. | Application show “Danh sách bệnh nhân” screen. |
| 2 | User touch on name of patient on list. | Application display Main screen with other patient. |

**Alternative Scenario:**

| Step | Action | Response |
|---|---|---|
| 2 | User touch “Back” button on device. | Application return previous screen. |

| |
|---|
| **Exceptions:**<br>N/A |
| **Business Rules:**<br>N/A |

#### 3.2.2.2.4 View patient details

**TABLE 3-7 Use case View patient details specification**

| USE-CASE VIEW PATIENT DETAILS SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-5 | **Use-case version** | 0.1 |
| **Use-case name** | View patient details | | |
| **Author** | LinhNC | | |
| **Date** | 28/01/2015 | **Priority** | High |
| **Actor:**<br>User | | | |
| **Summary:**<br>In this use-case, User view patient with detail information already have on application. | | | |
| **Goal:**<br>Application show patient's detail. | | | |
| **Triggers:**<br>User touch on view icon on list patient on "Danh sách bệnh nhân" screen. | | | |
| **Pre-conditions:**<br>User on "Danh sách bệnh nhân" screen. | | | |
| **Post-conditions:**<br>Patient detail be displayed on screen. | | | |

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch on button on patient list. | Application show "Chi tiết bệnh nhân" screen with patient's information. |

**Alternative Scenario:**

N/A

**Exceptions:**

N/A

**Business Rules:**

N/A

#### 3.2.2.2.5 Edit patient information

TABLE 3-8 Use case Edit patient information specification

| USE-CASE EDIT PATIENT INFORMATION SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-6 | **Use-case version** | 0.1 |
| **Use-case name** | Edit patient information | | |
| **Author** | LinhNC | | |
| **Date** | 28/01/2015 | **Priority** | High |
| **Actor:** User | | | |
| **Summary:** User updates a patient's information which already has on application. | | | |
| **Goal:** The patient's information is updated. | | | |
| **Triggers:** User touch on edit button on "Chi tiết bệnh nhân" screen. | | | |

**Pre-conditions:**

Application is showing “Chi tiết bệnh nhân” screen.

**Post-conditions:**

Application show “Chi tiết bệnh nhân” screen with patient’s information be updated.

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch on button on “Chi tiết bệnh nhân” screen. | Application show “Chỉnh sửa thông tin” screen with information of that patient. |
| 2 | User edits information into fields at the screens. | |
| 3 | User touch on “Lưu” button on “Chỉnh sửa thông tin” screen. | - Application check condition:<br>+ All field must be filled.<br>+ Patient name have maximum length is 30 and must not have special digit or number (@!~#$%^&*?<>:;)<br>- If User input incorrect information application show error notification (ER_001).<br>- If User input all correct information application save patient’s information to database and return“Chi tiết bệnh nhân” screen. |

**Alternative Scenario:**

| Step | Action | Response |
|---|---|---|
| 2, 3 | User touch “Nhập lại” button on “Chỉnh sửa thông tin” screen. | All information on fields at the screen be cleaned. |
| 2, 3 | User touch “Back” button on device. | Application show “Chi tiết bệnh nhân” screen. |

**Exceptions:**

| Step | Action | Response |
|---|---|---|
| 3 | User input information, touch on OK button. | ER_001: Show message error below the field which have incorrect information. |

**Business Rules:**

#### 3.2.2.2.6 Delete patient

**TABLE 3-9 Use case Delete patient specification**

| USE-CASE DELETE PATIENT SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-7 | **Use-case version** | 0.1 |
| **Use-case name** | Delete patient | | |
| **Author** | LinhNC | | |
| **Date** | 28/01/2015 | **Priority** | High |
| **Actor:** User | | | |
| **Summary:** User delete a patient which already has on application. | | | |
| **Goal:** The patient is deleted. | | | |
| **Triggers:** User touch on “Xóa” button on “Chi tiết bệnh nhân” screen. | | | |
| **Pre-conditions:** Application is showing “Chi tiết bệnh nhân” screen. | | | |
| **Post-conditions:** Application show “Danh sách bệnh nhân” screen. | | | |

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch on “Xóa” button on “Chi tiết bệnh nhân” screen | Application show dialog confirm. |
| 2 | User touch on “OK” button on dialog. | Application delete patient & related reminders from database and return “Danh sách bệnh nhân” screen. |

**Alternative Scenario:**

| Step | Action | Response |
|---|---|---|
| 2 | User touch “Hủy” button on dialog. | Application return “Chi tiết bệnh nhân” screen. |
| 2 | User touch “Back” button on device. | Application return “Chi tiết bệnh nhân” screen. |

**Exceptions:**

N/A

**Business Rules:**

N/A

#### 3.2.2.3 Health judgment

TABLE 3-10 Use case Health judgment specification

| USE-CASE HEALTH JUDGMENT SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-8 | **Use-case version** | 0.1 |
| **Use-case name** | Health judgment | | |
| **Author** | LinhNC | | |
| **Date** | 28/01/2015 | **Priority** | High |
| **Actor:** User | | | |
| **Summary:** User could enter information and application will return some common judgment about health situation of user. | | | |
| **Goal:** Give judgment of health situation. | | | |
| **Triggers:** User touch on “Đánh giá” button on Main screen or touch on “Đánh giá” button on Navigation drawer. | | | |
| **Pre-conditions:** User on Main screen or Navigation drawer. | | | |

**Post-conditions:**

Return some common judgment about health situation of user.

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | Touch on "Đánh giá" button on Main screen or Navigation drawer. | Application display "Đánh giá" screen |
| 2 | User choose index range from fields at the screens. | |
| 3 | User touch "OK" button on "Đánh giá" screen. | - If one of three index is filled, OK button is enabled.<br>- Application return some common judgments about health situation of user at judgment field. |

**Alternative Scenario:**

| Step | Action | Response |
|---|---|---|
| 2 | User touch "Nhập lại" button on "Đánh giá" screen. | All information on fields at the screen be cleaned. |
| 2 | User touch "Back" button on device. | Application show Main screen. |

**Exceptions:**

N/A

**Business Rules:**

N/A

#### 3.2.2.4 Recently status

TABLE 3-11 Use case Recently status specification

| USE-CASE RECENTLY STATUS SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-9 | **Use-case version** | 0.1 |
| **Use-case name** | Recently status | | |
| **Author** | LinhNC | | |
| **Date** | 28/01/2015 | **Priority** | High |

**Actor:**

User

**Summary:**

In this use-case, user view recently list.

**Goal:**

Application show list of judgments of current user.

**Triggers:**

User touch on "Tình trạng gần đây" button on Navigation drawer.

**Pre-conditions:**

User on Navigation drawer.

**Post-conditions:**

List index and judgments be displayed on screen.

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch on "Tình trạng gần đây" button on Navigation drawer. | Application show "Tình trạng gần đây" screen. |

**Alternative Scenario:**

N/A

**Exceptions:**

N/A

**Business Rules:**

**N/A**

#### 3.2.2.4.1 Recently status detail

TABLE 3-12 Use case Recently status detail specification

| USE-CASE RECENTLY STATUS DETAIL SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-10 | **Use-case version** | 0.1 |
| **Use-case name** | Recently status detail | | |

| | | | |
|---|---|---|---|
| **Author** | LinhNC | | |
| **Date** | 28/01/2015 | **Priority** | High |

**Actor:**

User

**Summary:**

In this use-case, user view index and judgments follow time.

**Goal:**

Application show index and judgments of current user.

**Triggers:**

User touch on an item date on recently status list.

**Pre-conditions:**

User on "Tình trạng gần đây" screen.

**Post-conditions:**

List index and judgments be displayed on screen.

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch on "Tình trạng gần đây" button on Navigation drawer. | Application show "Tình trạng gần đây" screen. |
| 2 | User touch on item date on list. | Application show "Chi tiết tình trạng" screen. |

**Alternative Scenario:**

N/A

**Exceptions:**

N/A

**Business Rules:**

N/A

### 3.2.2.5 Reminder

#### 3.2.2.5.1 View reminder list

TABLE 3-13 Use case View reminder list specification

| USE-CASE VIEW REMINDER LIST SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-11 | **Use-case version** | 0.1 |
| **Use-case name** | View reminder list | | |
| **Author** | LinhNC | | |
| **Date** | 23/01/2015 | **Priority** | High |
| **Actor:** User | | | |
| **Summary:** In this use-case, User view list reminders already have on application. | | | |
| **Goal:** Application show list reminder. | | | |
| **Triggers:** User touch on "Nhắc nhở uống thuốc" button on Main screen or touch on "Nhắc nhở uống thuốc" button on Navigation drawer. | | | |
| **Pre-conditions:** User on Main screen or Navigation drawer. | | | |
| **Post-conditions:** List reminder be displayed on screen. | | | |

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch "Nhắc nhở uống thuốc" button on Main screen or Navigation drawer. | Application show "Danh sách nhắc nhở" screen with list of all reminders related to current user on application. |
| 2 | User filter reminder list from Dropdown List on List Reminder Screen. | When patient choose All Reminder, the application will display all reminders which belongs to that patient , when patient |

| | |
|---|---|
| | choose Today's Reminder, just the reminders which will be alarmed today are displayed |
| **Alternative Scenario:** N/A | |
| **Exceptions:** N/A | |
| **Business Rules:** N/A | |

#### 3.2.2.5.2 Add new reminder

**TABLE 3-14 Use case Add new reminder specification**

| USE-CASE ADD NEW REMINDER SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-12 | **Use-case version** | 0.1 |
| **Use-case name** | Add new reminder | | |
| **Author** | LinhNC | | |
| **Date** | 23/01/2015 | **Priority** | High |
| **Actor:** User | | | |
| **Summary:** User add new reminder. | | | |
| **Goal:** Application add new reminder to database. | | | |
| **Triggers:** User touch on add button on "Danh sách nhắc nhở" screen. | | | |
| **Pre-conditions:** Application is showing "Danh sách nhắc nhở" screen. | | | |

**Post-conditions:**

Application show “Danh sách nhắc nhở” screen with new reminder.

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| **1** | User touch on Add button on “Danh sách nhắc nhở” screen. | Application show “Thêm lời nhắc nhở” screen. |
| **2** | User enter information into fields at the screens. | |
| **3** | User touch on “Lưu” button on “Thêm lời nhắc nhở” screen. | - Application check condition:<br>+ “Tên thuốc” must be filled and not have special character (@!~#$%^&*?<>:;).<br>+ “Số lượng” index must be filled and positive.<br>+ “Số thuốc/lần” index must be filled and positive.<br>+”Chu kì” index must be filled.<br>+”Giờ uống” index have at least one.<br>- If User input incorrect information application show error notification. (ER_001)<br>- If User input all correct information application add new reminder to database, remind to user on time and return “Danh sách nhắc nhở” screen. |

**Alternative Scenario:**

| Step | Action | Response |
|---|---|---|
| **2, 3** | User touch “Nhập lại” button on “Thêm lời nhắc nhở” screen. | All information on fields at the screen be cleaned. |
| **2, 3** | User touch “Back” button on device. | Application show “Danh sách nhắc nhở” screen. |

**Exceptions:**

| Step | Action | Response |
|---|---|---|
| **3** | Input information, touch on OK button | ER_001: Show message error below the field which have incorrect information. |

| **Business Rules:** |
| --- |
| N/A |

#### 3.2.2.5.3 Reminder details

**TABLE 3-15 Use case Reminder details specification**

| USE-CASE REMINDER DETAILS SPECIFICATION | | | |
| --- | --- | --- | --- |
| **Use-case no.** | UC-13 | **Use-case version** | 0.1 |
| **Use-case name** | Reminder details | | |
| **Author** | LinhNC | | |
| **Date** | 23/01/2015 | **Priority** | High |
| **Actor:** User | | | |
| **Summary:** In this use-case, User view reminders detail already have on application. | | | |
| **Goal:** Application show reminder detail**.** | | | |
| **Triggers:** User touch on reminder on list reminder on “Danh sách nhắc nhở” screen. | | | |
| **Pre-conditions:** User on “Danh sách nhắc nhở” screen. | | | |
| **Post-conditions:** Reminder detail be displayed on screen. | | | |

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch on a reminder on list reminder. | Application show "Chi tiết lời nhắc nhở" screen with reminder information. |

**Alternative Scenario:**

N/A

**Exceptions:**

N/A

**Business Rules:**

N/A

#### 3.2.2.5.4 Edit reminder

TABLE 3-16 Use case Edit reminder specification

| USE-CASE "THAY ĐỔI LỜI NHẮC NHỞ" SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-14 | **Use-case version** | 0.1 |
| **Use-case name** | "Thay đổi lời nhắc nhở" | | |
| **Author** | LinhNC | | |
| **Date** | 23/01/2015 | **Priority** | High |
| **Actor:** User | | | |
| **Summary:** User updates a reminder which already has on application. | | | |
| **Goal:** The reminder is updated. | | | |
| **Triggers:** User touch on "Thay đổi" button on "Chi tiết lời nhắc nhở" screen. | | | |

**Pre-conditions:**

Application is showing “Chi tiết lời nhắc nhở” screen.

**Post-conditions:**

Application show “Chi tiết lời nhắc nhở” screen with reminders information be updated.

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch on “Sửa” button on “Chi tiết nhắc nhở” screen. | Application show “Thay đổi lời nhắc nhở” screen with information of that reminder. |
| 2 | User edits information into fields at the screens. | |
| 3 | User touch on “Lưu” button on “Thay đổi lời nhắc nhở” screen. | - Application check condition:<br>+ “Tên thuốc” must be filled and not have special character (@!~#$%^&*?<>:;).<br>+ “Số lượng” index must be filled and positive.<br>+ “Số thuốc/lần” index must be filled and positive.<br>+”Chu kì” index must be filled.<br>+”Giờ uống” index have at least one.<br>- If User input incorrect information application show error. (ER_001)<br>- If User input all correct information application update reminder to database and return “Chi tiết lời nhắc nhở” screen. |

**Alternative Scenario:**

| Step | Action | Response |
|---|---|---|
| 2, 3 | User touch “Nhập lại” button “Thay đổi lời nhắc nhở” screen. | All information on fields at the screen be cleaned. |
| 2, 3 | User touch “Back” button on device. | Application show “Chi tiết lời nhắc nhở” screen. |

**Exceptions:**

| Step | Action | Response |
|---|---|---|
| 3 | User input information, touch on “Lưu” button. | ER_001: Show message error below the field which have incorrect information. |

**Business Rules:**

N/A

#### 3.2.2.5.5 Delete reminder

**TABLE 3-17 Use case Edit reminder specification**

| USE-CASE "XÓA NHẮC NHỞ" SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-15 | **Use-case version** | 0.1 |
| **Use-case name** | "Xóa nhắc nhở" | | |
| **Author** | LinhNC | | |
| **Date** | 23/01/2015 | **Priority** | High |
| **Actor:** User | | | |
| **Summary:** User deletes a reminder which already has on application. | | | |
| **Goal:** The reminder is deleted. | | | |
| **Triggers:** User touch on "Xóa" button on "Thay đổi lời nhắc nhở" screen. | | | |
| **Pre-conditions:** Application is showing "Thay đổi lời nhắc nhở" screen. | | | |
| **Post-conditions:** Application show "Danh sách nhắc nhở" screen. | | | |

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch on "Xóa" button on "Chi tiết lời nhắc nhở" screen | Application show dialog confirm. |
| 2 | User touch on "OK" button on dialog. | Application delete reminder from database and return "Danh sách nhắc nhở" screen. |

**Alternative Scenario:**

| Step | Action | Response |
|---|---|---|
| 2 | User touch “Hủy” button on dialog. | Application return “Chi tiết lời nhắc nhở” screen. |
| 2 | User touch “back” button on device. | Application return “Chi tiết lời nhắc nhở” screen. |

**Exceptions:**

N/A

**Business Rules:**

N/A

#### 3.2.2.6 Hospital

**TABLE 3-18 Use case Hospital specification**

| USE-CASE HOSPITAL SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-16 | **Use-case version** | 0.1 |
| **Use-case name** | Hospital | | |
| **Author** | LinhNC | | |
| **Date** | 23/01/2015 | **Priority** | High |

**Actor:**

User

**Summary:**

In this use-case, User view list hospital and its information in Hanoi.

**Goal:**

Application show list hospital and its information in Hanoi.

**Triggers:**

User touch on “Hiển thị bệnh viện” button on Main screen or touch on “Hiển thị bệnh viện” button on Navigation drawer.

**Pre-conditions:**

User on Main screen or Navigation drawer.

GPS is on.

**Post-conditions:**

List hospital and its information in Hanoi.

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch “Hiển thị bệnh viện” button on Main screen or Navigation drawer. | - If GPS is off, Application show dialog request turn it on.<br>- If GPS is on, Application show “Hiển thị bệnh viện” screen with list hospital and hospital’s information on a map. |
| 2 | User touch on map screen. | Application show map full screen. |

**Alternative Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User click “Hủy” button on dialog request turn GPS on. | Application return Main screen. |
| 1 | User touch any hospital icon on the map. | Application show this hospital’s information. |
| 2 | User touch on “Back” button on device. | Application show base “Hiển thị bệnh viện” screen. |

**Exceptions:**

N/A

**Business Rules:**

N/A

#### *3.2.2.7 Help*

**TABLE 3-19 Use case Help specification**

| USE-CASE HELP SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-17 | **Use-case version** | 0.1 |
| **Use-case name** | Help | | |
| **Author** | LinhNC | | |

| **Date** | 23/01/2015 | **Priority** | High |
|---|---|---|---|

**Actor:**

User

**Summary:**

In this use-case, User view advice list from developers.

**Goal:**

Application show help list.

**Triggers:**

User touch on "Trợ giúp" button on Navigation drawer.

**Pre-conditions:**

User on Navigation drawer.

**Post-conditions:**

Application show help list.

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| **1** | User touch "Trợ giúp" button on Navigation drawer. | Application show "Trợ giúp" screen. |

**Alternative Scenario:**

| Step | Action | Response |
|---|---|---|
| **2,3** | User touch on "Back" button on device. | Application return Main screen. |

**Exceptions:**

N/A

**Business Rules:**

N/A

#### 3.2.2.7.1 Help detail

**TABLE 3-20 Use case Help detail specification**

| USE-CASE HELP DETAIL SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-18 | **Use-case version** | 0.1 |
| **Use-case name** | Help detail | | |
| **Author** | LinhNC | | |
| **Date** | 23/01/2015 | **Priority** | High |

**Actor:**

User

**Summary:**

In this use-case, User view advice information.

**Goal:**

Application show help detail.

**Triggers:**

User touch on an item on "Trợ giúp" sreen.

**Pre-conditions:**

User on "Trợ giúp" sreen.

**Post-conditions:**

Application show help detail.

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch "Trợ giúp" button on Navigation drawer. | Application show "Trợ giúp" screen. |
| 2 | User tap on an item in help lis. | Application show "Chi tiết lời khuyên" screen. |

**Alternative Scenario:**

N/A

| |
|---|
| **Exceptions:**<br>N/A |
| **Business Rules:**<br>N/A |

### 3.2.2.8 About us

**TABLE 3-21 Use case Help detail specification**

| USE-CASE "VỀ CHÚNG TÔI" SPECIFICATION | | | |
|---|---|---|---|
| **Use-case no.** | UC-18 | **Use-case version** | 0.1 |
| **Use-case name** | "Về chúng tôi" | | |
| **Author** | LinhNC | | |
| **Date** | 23/01/2015 | **Priority** | High |
| **Actor:**<br>User | | | |
| **Summary:**<br>In this use-case, User view information about developers. | | | |
| **Goal:**<br>Application show information about developers. | | | |
| **Triggers:**<br>User touch on "Về chúng tôi" button on Navigation drawer. | | | |
| **Pre-conditions:**<br>User on Navigation drawer. | | | |
| **Post-conditions:**<br>Application show information about developers. | | | |

**Main success Scenario:**

| Step | Action | Response |
|---|---|---|
| 1 | User touch "Trợ giúp" button on Navigation drawer. | Application show "Trợ giúp" screen. |

| |
|---|
| **Alternative Scenario:**<br>N/A |
| **Exceptions:**<br>N/A |
| **Business Rules:**<br>N/A |

## 3.3 Non-Functional Requirements

### 3.3.1 Usability

- The interface should be elegant, simple and outstanding.
- Text, image text and help document should be in Vietnamese.
- A normal user can understand how to use less than 5 minutes.
- Styles, color and terminologies are consistent and exact.

### 3.3.2 Reliability

- Initial data should be collected carefully and correctly.
- There are no spelling errors.
- High accuracy.
- No conflict with other software.
- Bug or defect:

  + Minor bugs: bugs related to GUI.

  + Significant bugs: bugs related to minor business logic.

  + Critical bugs: function can't execute correctly.

### 3.3.3 Performance

- Every screen should be completely displayed within:
  - View in offline mode: Load time < 1 second
  - Connect Google map: < 3 second
- Application must consume least battery, RAM as possible depends on devices.

# 4 Software Design Specification

## 4.1 Architecture Representation

### 4.1.1 Overall System Architectural

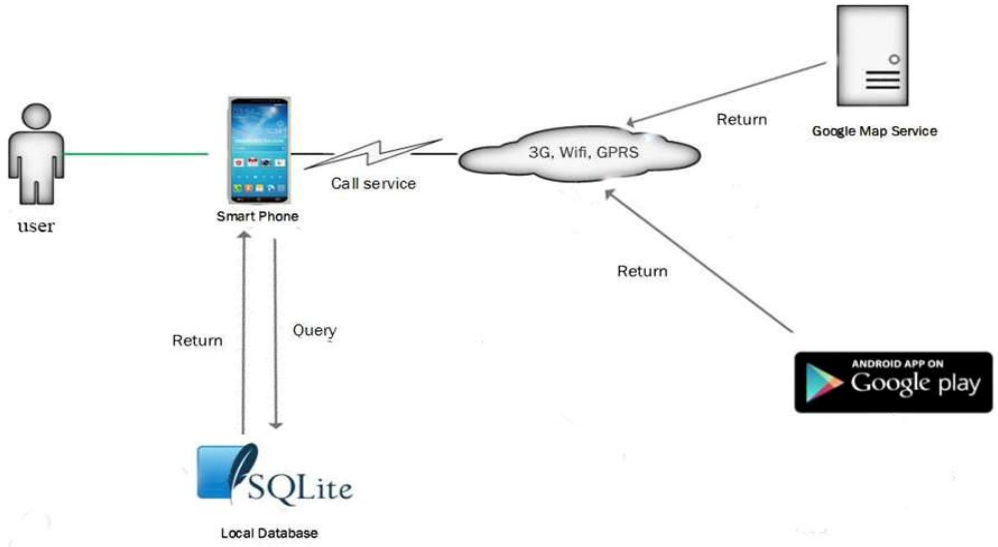



Figure 4-1 HCP system overall architecture

The HCP system have two main kinds of actors: User and Admin.

User downloads application from Google play store. User can insert or query data while using HCP's function, application will return results from SQLite database or save data into database. When new version of HCP which has updated or new functions as well as updated database is uploaded to play store, user can download and use latest version of HCP. Beside it, we also use Google map service to display map of Hospital function.

Admin take responsibility of update functions and data into database then upload it to Google play store. It can atomically download to user's device when it connect to the Internet.

### 4.1.2 Application Architecture

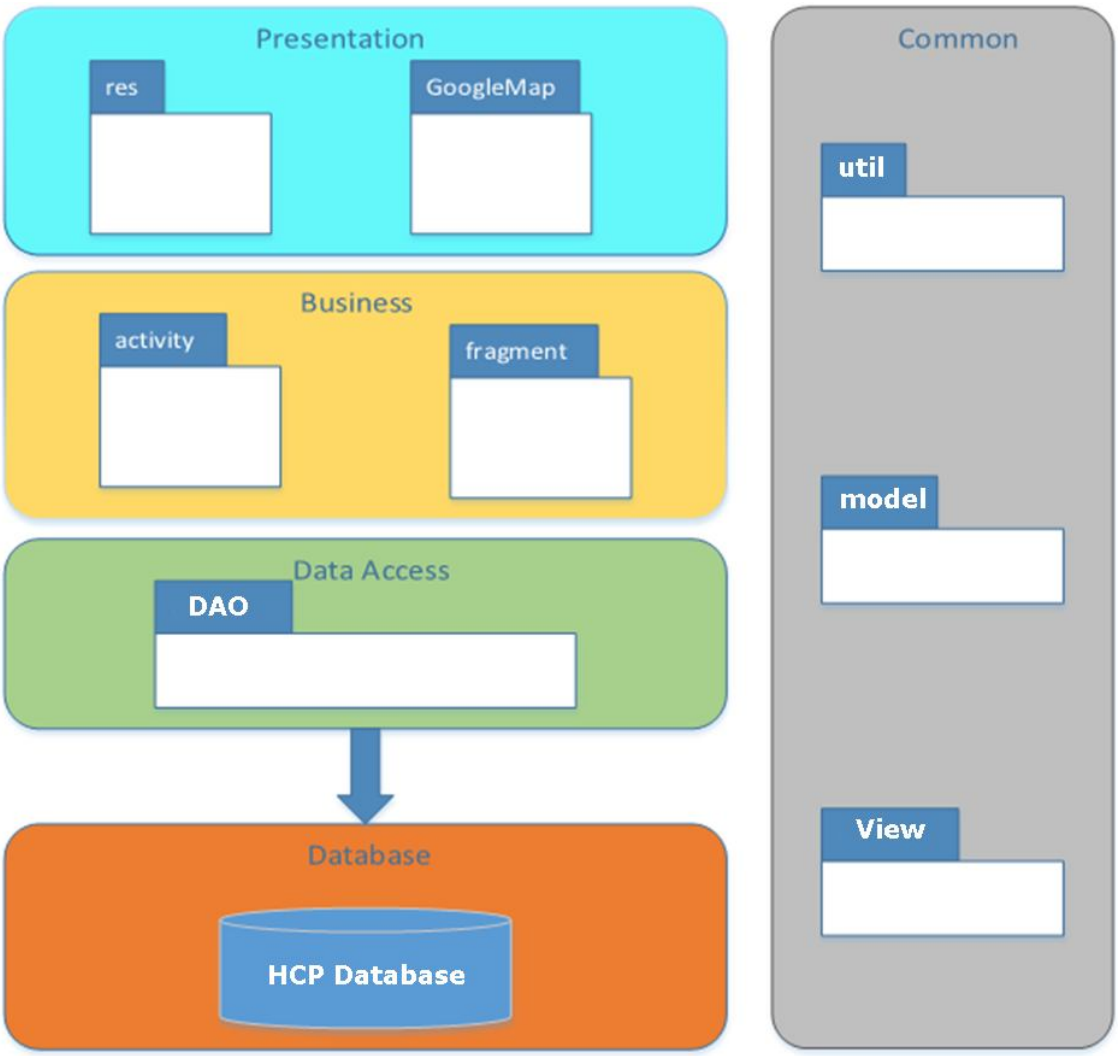



Figure 4-2 MVC model

HCP Main Application uses Model – View – Controller design pattern.

User actions in the view layer that create or modify data are communicated through a controller object and result in the creation or updating of a model object. When a model objects changes, it notifies a controller object, which updates the appropriate view objects.

### 4.1.3 Architectural Goals & Constraints

#### *4.1.3.1 Design and Implement Strategy*

- Basing on use case diagram to designing sequence diagram and then design activity diagram and class diagram.

- User Interface are designed based on Google's design style for Android.

#### 4.1.3.2 Design Tools

| No | Tool Name | Function |
|---|---|---|
| 1 | Enterprise Architect | Design sequence diagram, package diagram, class diagram |
| 2 | www.draw.io | Design ERD, screen flow |
| 3 | Jusinmind | Design GUI |
| 4 | Visual Paradgm | Design ERD |

## 4.2 Architecture Design

### 4.2.1 Logical view

#### 4.2.1.1 Patient

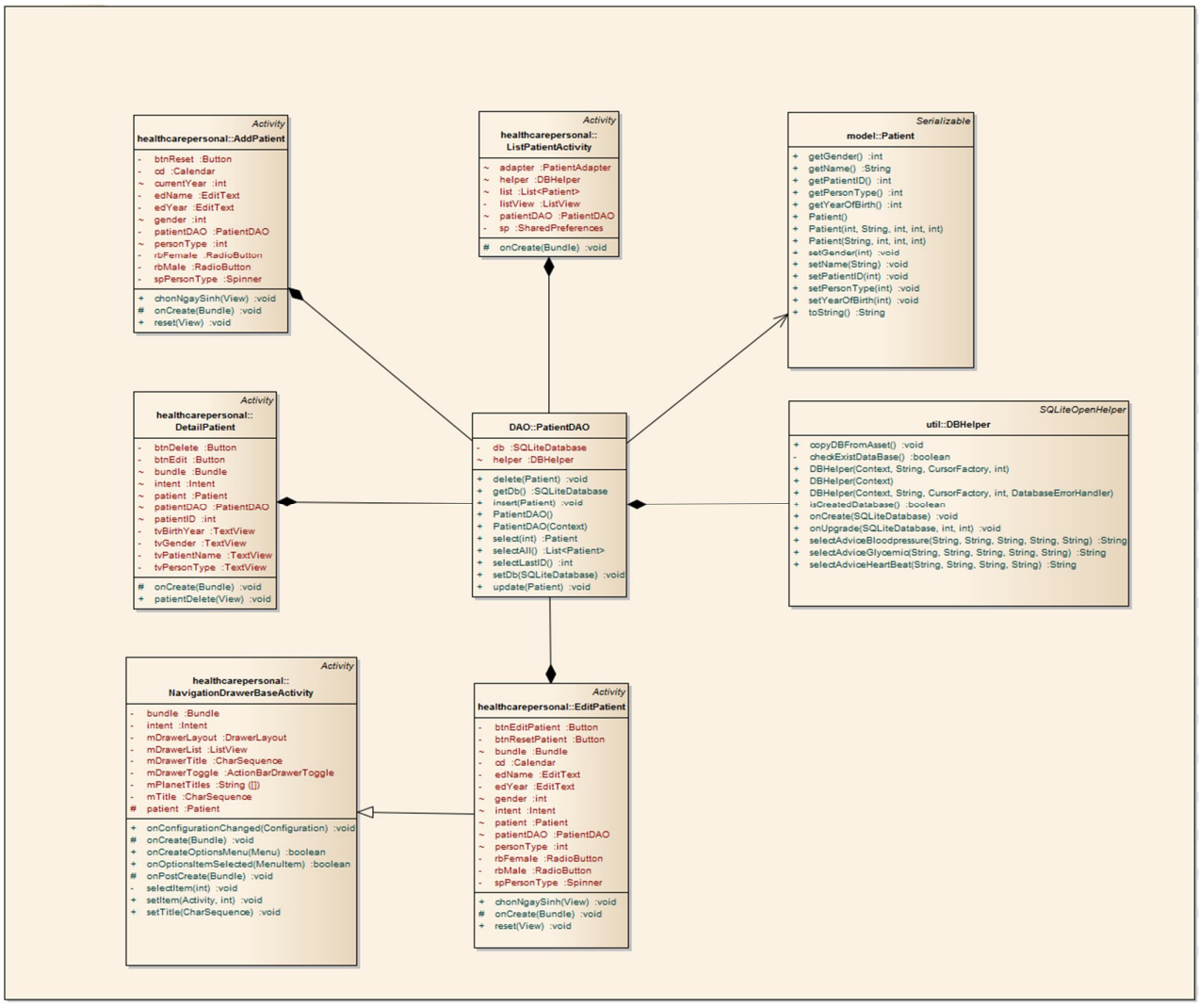



Figure 4-3 Patient diagram

#### 4.2.1.1.1 DBHelper

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| context | Context | private | The context which is needed |

❖ **Methods**

**Name**: selectAdviceGlycemic ()

- **Return:** String
- **Visibility:** public
- **Purpose:** select and return the advice about Glycemic base on the input index

**Parameters and return**: return the advice about Glycemic

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | status | String | In | The String express the condition of user (0 is Before Eating, 1 is After eating) |
| 02 | glycemic | String | In | The String contain the Glycemic index |
| 03 | gender | String | In | The String express the gender of user |
| 04 | age | String | In | The String express the age of user |
| 05 | personType | String | In | The String express the person type of user |

**Name**: selectAdviceBloodPressure ()

- **Return:** String
- **Visibility:** public
- **Purpose:** select and return the advice about Bloodpressure base on the input index

**Parameters and return**: return the advice about Glycemic

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | maxBloodPressure | String | In | The String contain the MaxBloodPressure index |
| 02 | minBloodPressure | String | In | The String contain the MinBloosPressure index |
| 03 | gender | String | In | The String express the gender of user |
| 04 | age | String | In | The String express the age of user |

| | | | | |
|---|---|---|---|---|
| 05 | personType | String | In | The String express the person type of user |

| **Name**: selectAdviceHeartBeat () | | | | |
|---|---|---|---|---|
| • **Return:** String<br>• **Visibility:** public<br>• **Purpose:** select and return the advice about HeartBeat base on the input index | | | | |
| **Parameters and return**: return the advice about HeartBeat | | | | |
| No | Parameter | Type | In/Out | Description |
| 01 | heartBeat | String | In | The String contain the HeartBeat index |
| 02 | gender | String | In | The String express the gender of user |
| 03 | age | String | In | The String express the age of user |
| 04 | personType | String | In | The String express the person type of user |

| **Name**: isCreatedDatabase () |
|---|
| • **Return:** boolean<br>• **Visibility:** public<br>• **Purpose:** check if the database was created or not |
| **Parameters and return**: return if the Database is created |

| **Name**: checkExistDatabase () |
|---|
| • **Return:** boolean<br>• **Visibility:** private<br>• **Purpose:** check if the database is already exist or not |
| **Parameters and return**: return if the database is already exist or not |

| **Name**: copyDBFromAsset () |
|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** copy database from asset into project |
| **Parameters and return**: none |

#### 4.2.1.1.2 PatientDAO

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| db | SQLiteDatabase | private | The database which this class will work with |
| helper | DBHelper | None | The class helps PatientDAO work with database |

❖ **Methods**

| **Name**: insert() | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Insert the information of inputted Patient into database | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | Patient | Patient | In | The Patient to insert into database |

| **Name**: delete() | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Remove the inputted Patient from database | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | Patient | Patient | In | The Patient need to be removed from database |

| **Name**: select() | | | | |
|---|---|---|---|---|
| • **Return:** Patient<br>• **Visibility:** public<br>• **Purpose:** Get and return the Patient by the Id of that patient | | | | |
| **Parameters and return**: return Patient which has the Id equal patientID | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | patientID | int | In | The Id of the Patient which need to be gotten |

| **Name**: selectAll() |
|---|
| • **Return:** List<Patient><br>• **Visibility:** public<br>• **Purpose:** Get and return the List of all Patient in the database |
| **Parameters and return**: return the List of all Patient |

| **Name**: selectLastID() |
|---|
| • **Return:** int<br>• **Visibility:** public<br>• **Purpose:** Get and return the ID of the last record of Patient in database |
| **Parameters and return**: return the ID of the last record of Patient |

| **Name**: update() | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Update the information of an exist Patient with inputted information | | | | |
| **Parameters and return**: return Patient which has the Id equal patientID | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | patient | Patient | In | The Patient contain the information to be update to the record of Patient which has the same ID |

#### 4.2.1.1.3 Patient

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| patientID | int | private | The Id for each Patient record in datatbase |
| name | String | private | The Name of the patient |
| yearOfBirth | int | private | The Year of birth of Patient |
| gender | int | private | The gender of Patient (0 is “Nam”, 1 is “Nữ”) |
| personType | int | private | The person type of Patient (0 is “Bình thường” ,1 is “Hay vận động”) |

#### 4.2.1.1.4 AddPatient

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| btnReset | Button | private | The button to reset all Edit Text, radio button and spinner to default value |
| cd | Calendar | private | The Calendar's instance to get [today] value |
| currentYear | Int | private | The int present the current year |
| edName | EditText | private | The Edit text receive patient's name |
| edYear | EditText | private | The Edit text receive patient's Year of Birth |
| gender | Int | private | The int present for the gender of Patient ( 0 is "Nam", 1 is "Nữ") |
| patientDAO | PatientDAO | private | The PatientDAO for the activity to work with database |
| personType | Int | private | The int present for the person Type of Patient ( 0 is "Bình thường", 1 is "Hay vận động") |
| rbFemale | RadioButton | private | The radio button present the "Nam" choice |
| rbMale | RadioButton | private | The radio button present the "Nữ" choice |
| spPersonType | Spinner | private | The Spinner contain choices for Person Type ("Bình thường"/ "Hay vận động") |

❖ **Methods**

**Name**: chonNgaySinh ()

- **Return:** void
- **Visibility:** public
- **Purpose:** Set the current Year for the number picker which will appear when user tab on Edit Text of the Year Of Birth

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | view | View | In | The View which is needed |

**Name**: reset ()

- **Return:** void

- **Visibility:** public
- **Purpose:** Reset all the edit text, radio button and spinner to default values

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | view | View | In | The View which is needed |

#### 4.2.1.1.5 DetailPatient

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| btnDelete | Button | private | The button to delete a Patient |
| btnEdit | Button | private | The button to pass to the Edit Patient screen |
| bundle | Bundle | private | The bundle which need to get the variable pass from the last screen |
| intent | Intent | private | The Intent which need to get the variable pass from the last screen |
| patient | Patient | private | The patient's instance to get the patient pass from the last screen |
| patientDAO | PatientDAO | private | The PatientDAO to work with database |
| patientID | int | private | The Is of the patient |
| tvBirthYear | TextView | private | The TextView to display patient's Year Of Birth |
| tvPatientName | TextView | private | The TextView to display patient's Name |
| tvPersonType | TextView | private | The TextView to display patient's Person Type |
| tvGender | TextView | private | The TextView to display patient's Gender |

❖ **Methods**

**Name**: patientDelete ()

- **Return:** none
- **Visibility:** public
- **Purpose:** Display a dialog with 2 choices : "Xóa" and "Hủy". The Patient will be remove from database when user choose "Xóa", or return to Detail Patient screen when user choose "Hủy"

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 01 | view | View | In | The View which is needed |

#### 4.2.1.1.6 EditPatient

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| btnEditPatient | Button | private | The button which will do the work update the patient with the inputted informations |
| btnResetPatient | Button | private | The Button which will reset all the field on the screen to the values before editting |
| bundle | Bundle | private | The bundle which need to get the variable pass from the last screen |
| intent | Intent | private | The Intent which need to get the variable pass from the last screen |
| patient | Patient | private | The patient's instance to get the patient pass from the last screen |
| Cd | Calendar | private | The Calendar's instance to get [today] value |
| patientDAO | PatientDAO | private | The PatientDAO to work with database |
| personType | Int | private | The int present for the person Type of Patient ( 0 is "Bình thường", 1 is "Hay vận động") |
| rbFemale | RadioButton | private | The radio button present the "Nam" choice |
| rbMale | RadioButton | private | The radio button present the "Nữ" choice |
| spPersonType | Spinner | private | The Spinner contain choices for Person Type ("Bình thường"/ "Hay vận động") |
| edName | EditText | private | The Edit text receive patient's name |
| edYear | EditText | private | The Edit text receive patient's Year of Birth |
| gender | Int | private | The int present for the gender of Patient ( 0 is "Nam", 1 is "Nữ") |

❖ **Methods**

**Name**: chonNgaySinh ()

- **Return:** void
- **Visibility:** public

| |
|---|
| • **Purpose:** Set the current Year for the number picker which will appear when user tab on Edit Text of the Year Of Birth |
| **Parameters and return**: |

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | view | View | In | The View which is needed |

| **Name**: reset () |
|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Reset all the edit text, radio button and spinner to default values |
| **Parameters and return**: |

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | view | View | In | The View which is needed |

#### 4.2.1.1.7 ListPatientActivity

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| adapter | PatientAdapter | private | The PatientAdapter to help display the Patient's List with customized ListView |
| helper | DBHelper | private | The DBHelper to connect with datatbase |
| list | List<Patient> | private | The List contain all patient |
| listView | ListView | private | The ListView to display Patient's List |
| patientDAO | PatientDAO | private | The PatientDAO to work with database |
| sp | SharedPreference | private | The SharedPreference which needed |

#### 4.2.1.1.8 NavigatorDrawerBaseActivity

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| bundle | Bundle | private | The bundle which need to get the variable pass from the last screen |
| intent | Intent | private | The Intent which need to get the variable pass from the last screen |
| patient | Patient | protected | The patient's instance to get the patient pass from the last screen |

| | | | |
|---|---|---|---|
| mDrawerLayout | DrawerLayout | private | The DrawerLayout contain all element of navigator drawer |
| mDrawerList | ListView | private | The ListView contain all context in navigator |
| mDrawerTitle | CharSequence | private | The String display title of navigator |
| mDrawerToggle | ActionBarDrawerToggle | private | The ActionBarDrawerToggle make button toggle on navigator enable in action bar |
| mPlanetTitles | String{[]} | private | String Array contain all name of item in navigator |

❖ **Methods**

**Name**: setItem()

- **Return: void**
- **Visibility: public**
- **Purpose: change the current layout of screen to frame layout**

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | `activity` | `Activity` | In | Define which Activity transform to frame layout |
| 02 | `viewID` | `int` | In | Define which screen ID in R class |

**Name**: selectItem ()

- **Return: void**
- **Visibility: private**
- **Purpose: enable action when click an item in navigator**

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | `position` | `int` | In | Define position of item in navigator |

**Name**: setTitle ()

- **Return: void**
- **Visibility: public**
- **Purpose: set title of navigator**

| **Parameters and return**: | | | | |
|---|---|---|---|---|
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | `title` | `CharSequence` | In | Set name for navigator |

### 4.2.1.2 Reminder

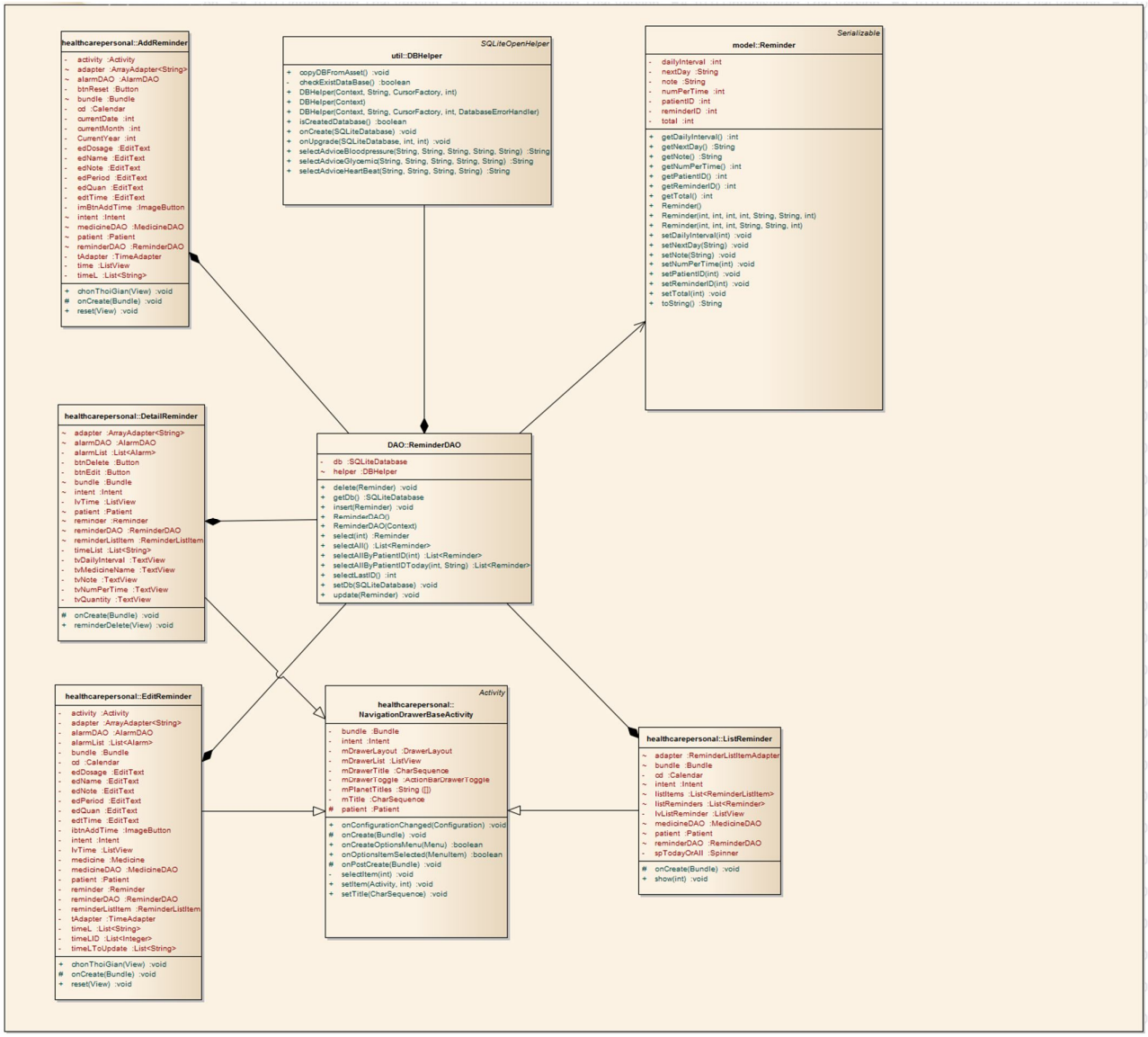



Figure 4-4 Reminder diagram

#### 4.2.1.2.1 ReminderDAO

- **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| db | SQLiteDatabase | private | The database which this class will work with |

| | | | |
|---|---|---|---|
| helper | DBHelper | None | The class helps HospitalDAO work with database |

❖ **Methods**

| **Name**: insert () | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Insert the information of inputted Reminder into database | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | reminder | Reminder | In | The Reminder to insert into database |

| **Name**: update() | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Update the information of an exist Reminder with inputted information | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | reminder | Reminder | In | The Reminder contain the information to be update to the record of Reminder which has the same ID |

| **Name**: selectAllByPatientID () | | | | |
|---|---|---|---|---|
| • **Return:** List< Reminder ><br>• **Visibility:** public<br>• **Purpose:** Get and return the List of all Reminder which belongs to the Patient who has Id equal patientID in the database | | | | |
| **Parameters and return**: return List of Medicine | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | patientID | int | In | The Id of the Patient which reminders belong to |

| **Name**: select () |
|---|

- **Return:** Reminder
- **Visibility:** public
- **Purpose:** Get and return the Reminder by the Id of that Reminder

**Parameters and return**: return Reminder which has the Id equal id

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | reminderID | int | In | The Id of the Reminder which need to be gotten |

**Name**: selectAllByPatientIDToday ()

- **Return:** List< Reminder >
- **Visibility:** public
- **Purpose:** Get and return the List of all Reminder which will be alarmed at that dayand belongs to the Patient who has Id equal patientID in the database

**Parameters and return**: return Reminder List

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | today | String | In | The String presents the day the mothod called |
| 02 | patientID | int | In | The Id of the Patient which reminders belong to |

**Name**: delete()

- **Return:** void
- **Visibility:** public
- **Purpose:** Remove the inputted Reminder from database

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | reminder | Reminder | In | The Reminder need to be removed from database |

**Name**: selectAll ()

- **Return:** List< Reminder >
- **Visibility:** public
- **Purpose:** Get and return the List of all Reminder in the database

**Parameters and return**: return the List of all Reminder

| **Name**: selectLastID () |
|---|
| • **Return:** int<br>• **Visibility:** public<br>• **Purpose:** Get and return the ID of the last record of Reminder in database |
| **Parameters and return**: return the ID of the last record of Reminder |

#### 4.2.1.2.2 Reminder

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| reminderID | int | private | The Id for each Reminder record in database |
| patientID | int | private | The Id of the Patient who the Reminder belongs to |
| numPerTime | int | private | The number of tablet for patient to take each time |
| dailyInterval | int | private | The number of day between the time Patient take the medicine |
| nextDay | String | private | The Next Day patient take the medicine |
| total | int | private | The total number of tablet patient need to take os the medicine |
| note | String | private | Some bonus information for the medicine |

#### 4.2.1.2.3 AddReminder

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| activity | Activity | private | The Activity which is needed |
| adapter | ArrayAdapter<String> | private | The adapter to help display the time list |
| btnReset | Button | private | The button to reset all Edit Text to default value |
| cd | Calendar | private | The Calendar's instance to get [today] value |
| currentYear | Int | private | The int present the current year |
| currentMonth | Int | private | The int present the current month |
| currentDate | Int | private | The int present the current day |
| bundle | Bundle | private | The bundle which need to get the |

| | | | |
|---|---|---|---|
| | | | variable pass from the last screen |
| intent | Intent | private | The Intent which need to get the variable pass from the last screen |
| patient | Patient | private | The patient's instance to get the patient pass from the last screen |
| edDosage | EditText | private | The EditText to input the information of Dosage |
| edName | EditText | private | The EditText to input the information of Medicine's Name |
| edNote | EditText | private | The EditText to input the information of Note |
| edPeriod | EditText | private | The EditText to input the information of Period |
| edQuan | EditText | private | The EditText to input the information of Quantity |
| edtTime | EditText | private | The EditText to input the information of Time |
| imBtnAddTime | ImageButton | private | The ImageButton to add a time to Time ListView |
| medicineDAO | MedicineDAO | private | The MedicineDAO to help work with database about Medicine |
| alarmDAO | AlarmDAO | private | The AlarmDAO to help work with database about Alarm |
| reminderDAO | ReminderDAO | private | The ReminderDAO to help work with database about Reminder |
| tAdapter | TimeAdapter | private | The TimeAdapter to help display the time list |
| time | ListView | private | The ListView to display the time list |
| timeL | List<String> | private | The List of time |

❖ **Methods**

| **Name**: chonThoiGian () | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Set the current Date for the date picker which will appear when user tab on Edit Text of the Time | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |

| | | | | |
|---|---|---|---|---|
| 01 | view | View | In | The View which is needed |

| **Name**: reset () | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Reset all the edit text to default values | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | view | View | In | The View which is needed |

#### 4.2.1.2.4 ListReminder

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| cd | Calendar | private | The Calendar's instance to get [today] value |
| bundle | Bundle | private | The bundle which need to get the variable pass from the last screen |
| intent | Intent | private | The Intent which need to get the variable pass from the last screen |
| patient | Patient | private | The patient's instance to get the patient pass from the last screen |
| medicineDAO | MedicineDAO | private | The MedicineDAO to help work with database about Medicine |
| reminderDAO | ReminderDAO | private | The ReminderDAO to help work with database about Reminder |
| adapter | ReminderListItemAdapter | private | The adapter to help display the reminder list with customized ListView |
| listItems | List<ReminderListItem> | private | The List of ReminderListItem |
| listReminders | List<Reminder> | private | The List of Reminder |
| lvListReminder | ListView | private | The ListView which will display the ReminderListItem's List |
| spTodayOrAll | Sprinner | private | The Spinner for user to choose how to display the ReminderListItem's List ("Đơn thuốc hôm nay" / "Tất cả đơn thuốc") |

❖ **Methods**

| **Name**: show() | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Depend on the selection of user on the Spinner, the ReminderListItem's List will be display | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | view | View | In | The View which is needed |

#### 4.2.1.2.5 DetailReminder

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| adapter | ArrayAdapter<String> | private | The adapter to help display the time list |
| bundle | Bundle | private | The bundle which need to get the variable pass from the last screen |
| intent | Intent | private | The Intent which need to get the variable pass from the last screen |
| patient | Patient | private | The patient's instance to get the patient pass from the last screen |
| alarmDAO | AlarmDAO | private | The AlarmDAO to help work with database about Alarm |
| reminderDAO | ReminderDAO | private | The ReminderDAO to help work with database about Reminder |
| alarmList | List<Alarm> | private | The List of Alarm |
| btnDelete | Button | private | The button to display the dialog to delete Reminder |
| btnEdit | Button | private | The Button to pass to EditReminder screen |
| lvTime | ListView | private | The ListView to display the List of Time |
| reminder | Reminder | private | The reminder which will be display in this screen |
| reminderListItem | ReminderListItem | private | The ReminderListItem has been passed from the last screen |
| timelist | List<String> | private | The List of time |
| tvDailyInterval | TextView | private | The TextView to display the information of Period |
| tvMedicineName | TextView | private | The TextView to display the information of Medicine's Name |
| tvNote | TextView | private | The TextView to display the |

| | | | |
|---|---|---|---|
| | | | information of Note |
| tvNumPerTime | TextView | private | The TextView to display the information of Dosage |
| tvQuantity | TextView | private | The TextView to display the information of Quantity |

❖ **Methods**

| **Name**: reminderDelete () | | | | |
|---|---|---|---|---|
| • **Return:** none<br>• **Visibility:** public<br>• **Purpose:** Display a dialog with 2 choices : "Xóa" and "Hủy". The Reminder will be remove from database when user choose "Xóa", or return to Detail Reminder screen when user choose "Hủy" | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | view | View | In | The View which is needed |

#### 4.2.1.2.6 EditReminder

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| bundle | Bundle | private | The bundle which need to get the variable pass from the last screen |
| intent | Intent | private | The Intent which need to get the variable pass from the last screen |
| patient | Patient | private | The patient's instance to get the patient pass from the last screen |
| Cd | Calendar | private | The Calendar's instance to get [today] value |
| alarmDAO | AlarmDAO | private | The AlarmDAO to help work with database about Alarm |
| reminderDAO | ReminderDAO | private | The ReminderDAO to help work with database about Reminder |
| medicineDAO | MedicineDAO | private | The MedicineDAO to help work with database about Medicine |
| activity | Activity | private | The Activity which is needed |
| adapter | ArrayAdapter<String> | private | The adapter to help display the time list |
| edDosage | EditText | private | The EditText to input the information of Dosage |

| | | | |
|---|---|---|---|
| edName | EditText | private | The EditText to input the information of Medicine's Name |
| edNote | EditText | private | The EditText to input the information of Note |
| edPeriod | EditText | private | The EditText to input the information of Period |
| edQuan | EditText | private | The EditText to input the information of Quantity |
| edtTime | EditText | private | The EditText to input the information of Time |
| imBtnAddTime | ImageButton | private | The ImageButton to add a time to Time ListView |
| alarmList | List<Alarm> | private | The List of Alarm |
| lvTime | ListView | private | The ListView to display the List of Time |
| medicine | Medicine | private | The Medicine belongs to the Reminder which will be displayed in this screen |
| reminder | Reminder | private | The Reminder which will be displayed in this screen |
| reminderListItem | ReminderListItem | private | The ReminderListItem is passed from the last screen |
| tAdapter | TimeAdapter | private | The TimeAdapter to help display the List of Time by the customized ListView |
| timeL | List<String> | private | The list of Time |
| timeLID | List<Integer> | private | The List of ID of Alarm |
| timeLToUpdate | List<String> | private | The List of Time which will be use to update to database |

❖ **Methods**

**Name**: chonThoiGian ()

- **Return:** void
- **Visibility:** public
- **Purpose:** Set the current Date for the date picker which will appear when user tab on Edit Text of the Time

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | view | View | In | The View which is needed |

| **Name**: reset () | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Reset all the edit text to default values | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | view | View | In | The View which is needed |

### 4.2.1.3 Judgement

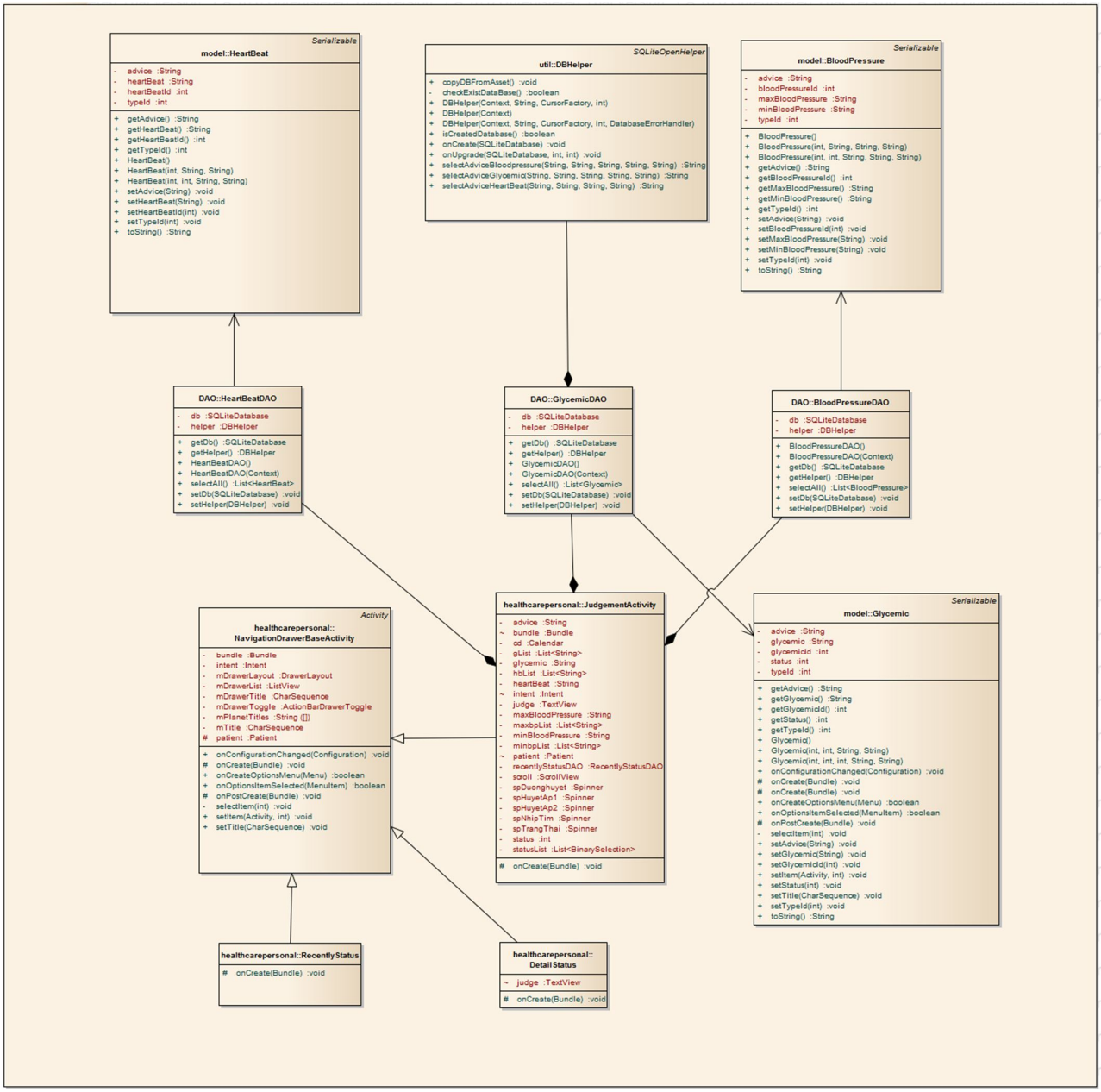


Figure 4-5 Judgement diagram

#### 4.2.1.3.1 RecentlyStatusDAO

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| db | SQLiteDatabase | private | The database which this class will work with |
| helper | DBHelper | None | The class helps RecentlyStatusDAO work with database |

❖ **Methods**

| **Name**: insert () | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Insert the information of inputted RecentlyStatus into database | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | recentlyStatus | RecentlyStatus | In | The RecentlyStatus to insert into database |

| **Name**: delete () | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Remove the inputted RecentlyStatus from database | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | recentlyStatus | RecentlyStatus | In | The RecentlyStatus need to be removed from database |

| **Name**: selectByPatientID () |
|---|
| • **Return:** List< RecentlyStatus ><br>• **Visibility:** public<br>• **Purpose:** Get and return the List of all RecentlyStatus in the database which belong to the Patient who has Id equal id |
| **Parameters and return**: return the List of all Patient |

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | id | int | In | The ID of the Patient which the reminders belong to |

#### 4.2.1.3.2 BloodPressureDAO

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| db | SQLiteDatabase | private | The database which this class will work with |
| helper | DBHelper | None | The class helps HospitalDAO work with database |

❖ **Methods**

| **Name**: selectAll() |
|---|
| • **Return:** List< BloodPressure ><br>• **Visibility:** public<br>• **Purpose:** Get and return the List of all BloodPressure in the database |
| **Parameters and return**: return the List of BloodPressure |

#### 4.2.1.3.3 GlycemicDAO

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| db | SQLiteDatabase | private | The database which this class will work with |
| helper | DBHelper | None | The class helps HospitalDAO work with database |

❖ **Methods**

| **Name**: selectAll() |
|---|
| • **Return:** List< Glycemic ><br>• **Visibility:** public<br>• **Purpose:** Get and return the List of all Glycemic in the database |
| **Parameters and return**: return the List of Glycemic |

#### 4.2.1.3.4 HeartBeatDAO

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| db | SQLiteDatabase | private | The database which this class will work with |
| helper | DBHelper | None | The class helps HospitalDAO work with database |

❖ **Methods**

| **Name**: selectAll() |
|---|
| • **Return:** List< HeartBeat ><br>• **Visibility:** public<br>• **Purpose:** Get and return the List of all HeartBeat in the database |
| **Parameters and return**: return the List of HeartBeat |

#### 4.2.1.3.5 Glycemic

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| glycemicID | int | private | The Id for each Glycemic record in database |
| glycemic | String | private | The String persent for each Glycemic index |
| typeId | int | private | The ID of the type for each Glycemic record |
| status | int | private | The status of person (Before Eating/ After Eating) |
| advice | String | private | The string contain the Advice |

#### 4.2.1.3.6 HeartBeat

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| heartBeatID | int | private | The Id for each HeartBeat record in database |
| typeId | int | private | The ID of the type for each HeartBeat record |
| heartBeat | String | private | The String persent for each HeartBeat index |

| | | | |
|---|---|---|---|
| advice | String | private | The string contain the Advice |

#### 4.2.1.3.7 BloosPressure

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| bloodPressureId | int | private | The Id for each BloodPressure record in database |
| typeId | int | private | The ID of the type for each BloodPressure record |
| maxBloodPressure | String | private | The String persent for each MaxBloodPressure index |
| minBloodPressure | String | private | The String persent for each MinBloodPressure index |
| advice | String | private | The string contain the Advice |

#### 4.2.1.3.8 RecentlyStatus

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| bundle | Bundle | private | The bundle which need to get the variable pass from the last screen |
| intent | Intent | private | The Intent which need to get the variable pass from the last screen |
| patient | Patient | private | The patient's instance to get the patient pass from the last screen |
| adapter | RecentlyStatusAdapter | private | The RecentlyStatusAdapter to help ListView to display the List of RecentlyStatus in customized way |
| recentlyStatusDAO | RecentlyStatusDAO | private | The RecentlyStatusDAO to work with database about RecentlyStatus |

| | | | |
|---|---|---|---|
| rsList | List<fu.capstone.healthcarepersonal.model.RecentlyStatus> | private | The list of RecentlyStatus |
| lvRS | ListView | private | The ListView to display The list of RecentlyStatus |

#### 4.2.1.3.9 DetailStatus

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| bundle | Bundle | private | The bundle which need to get the variable pass from the last screen |
| intent | Intent | private | The Intent which need to get the variable pass from the last screen |
| patient | Patient | private | The patient's instance to get the patient pass from the last screen |
| recentlyStatus | RecentlyStatus | private | The RecentlyStatus to display in this screen |
| tvDateTime | TextView | private | The TextView to display the information of the Date and time when the status was inputted |
| tvMaxBloodPressure | TextView | private | The TextView to display the information of MaxBloodPressure |
| tvMinBloodPressure | TextView | private | The TextView to display the information of MinBloodPressure |
| tvGlycemic | TextView | private | The TextView to display the information of Glycemic |
| tvStatus | TextView | private | The TextView to display the information of Status |
| tvHeartBeat | TextView | private | The TextView to display the information of HeartBeat |
| tvAdvice | TextView | private | The TextView to display the information of Advice |

#### 4.2.1.3.10 JudgementActivity

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| bundle | Bundle | private | The bundle which need to get the variable pass from the last screen |
| intent | Intent | private | The Intent which need to get the |

| | | | |
|---|---|---|---|
| | | | variable pass from the last screen |
| patient | Patient | private | The patient's instance to get the patient pass from the last screen |
| scroll | ScrollView | private | The ScrollView to TextView to be able to be scroll up and down when the content it out of the screen |
| cd | Calendar | private | The Calendar's instance to get [today] value |
| gList | List<String> | private | The List of the Glycemic index |
| hbList | List<String> | private | The List of the HeartBeat index |
| maxbpList | List<String> | private | The List of the MaxBloodPressure index |
| minbpList | List<String> | private | The List of the MinBloodPressure index |
| glycemic | String | private | The String to receive the value of Glycemic which is chosen |
| heartbeat | String | private | The String to receive the value of HeartBeat which is chosen |
| maxBloodPressure | String | private | The String to receive the value of MaxBloodPressure which is chosen |
| minBloodPressure | String | private | The String to receive the value of MinBloodPressure which is chosen |
| advice | String | private | The String to contain the value of Advice which is give base on the chosen index |
| judge | TextView | private | The TextView will display the advice |
| recentlyStatusDAO | RecentlyStatusDAO | private | The RecentlyStatusDAO to help the function work with database about RecentlyStatus |
| spDuongHuyet | Spinner | private | The Spinner to display the choice for Glycemic index |
| spHuyetAp1 | Spinner | private | The Spinner to display the choice for MaxBloodPressure index |
| spHuyetAp2 | Spinner | private | The Spinner to display the choice for MinBloodPressure index |
| spNhipTim | Spinner | private | The Spinner to display the choice for HeartBeat index |
| spTrangThai | Spinner | private | The Spinner to display the choice for Status |
| status | int | private | The number represent for the selection for the status (0 is "Trước ăn", 1 is "Sau ăn") |
| statusList | List<BinarySelection> | private | The List of Selection for Status |

#### 4.2.1.4 Map

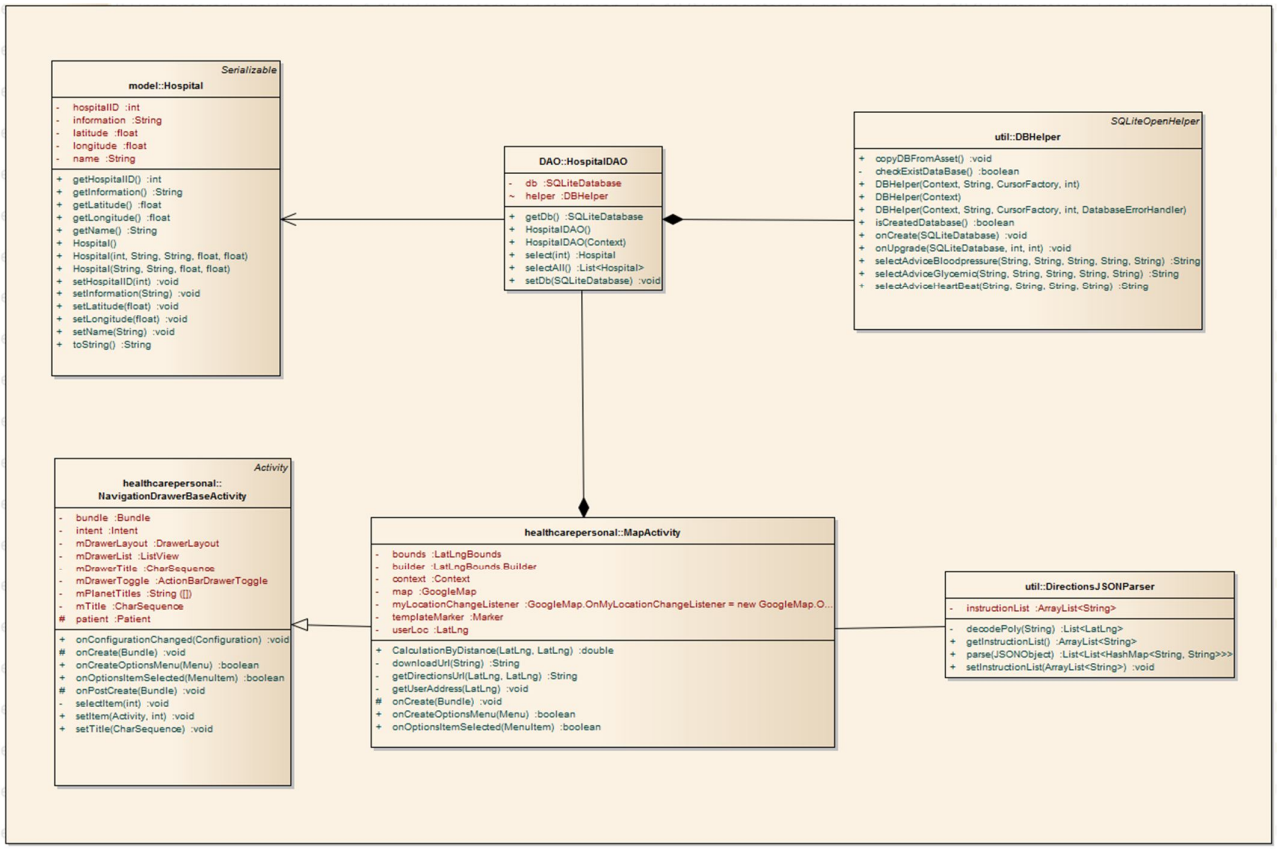



Figure 4-6 Map diagram

##### 4.2.1.4.1 HospitalDAO

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| db | SQLiteDatabase | private | The database which this class will work with |
| helper | DBHelper | None | The class helps HospitalDAO work with database |

❖ **Methods**

| **Name**: select () | | | | |
|---|---|---|---|---|
| • **Return:** Hospital<br>• **Visibility:** public<br>• **Purpose:** Get and return the Hospital which need to be displayed | | | | |
| **Parameters and return**: return Hospital which has the Id equal id | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |

| | | | | |
|---|---|---|---|---|
| 01 | id | int | In | The Id of the Hospital which need to be displayed |

| **Name**: selectAll () |
|---|
| • **Return:** List< Hospital ><br>• **Visibility:** public<br>• **Purpose:** Get and return the List of all Hospital in the database |
| **Parameters and return**: return the List of all Hospital |

#### 4.2.1.4.2 Hospital

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| hospitalID | int | private | The Id for each Hospital record in database |
| name | String | private | The Name of the hospital |
| information | String | private | Some basic information about the Hospital |
| longitude | float | private | The longitude of Hospital's address |
| latitude | float | private | The latitude of Hospital's address |

#### 4.2.1.4.3 MapActivity

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| bound | LatLngBounds | private | |
| builder | LatLngBoundsBuilder | private | |
| context | Context | private | |
| map | GoogleMap | private | |
| myLocationChangListener | GoogleMap.OnMyLocationChangeListener | private | |
| templateMarker | Marker | private | |
| userLoc | LatLng | private | |

❖ **Methods**

| **Name**: Hospital() |
|---|
| • **Return: Hospital**<br>• **Visibility: public** |

| • **Purpose:** Get and return the object Hospital | | | | |
|---|---|---|---|---|
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | 01 | int | In | All information of object Hospital |

#### 4.2.1.5 Advisory

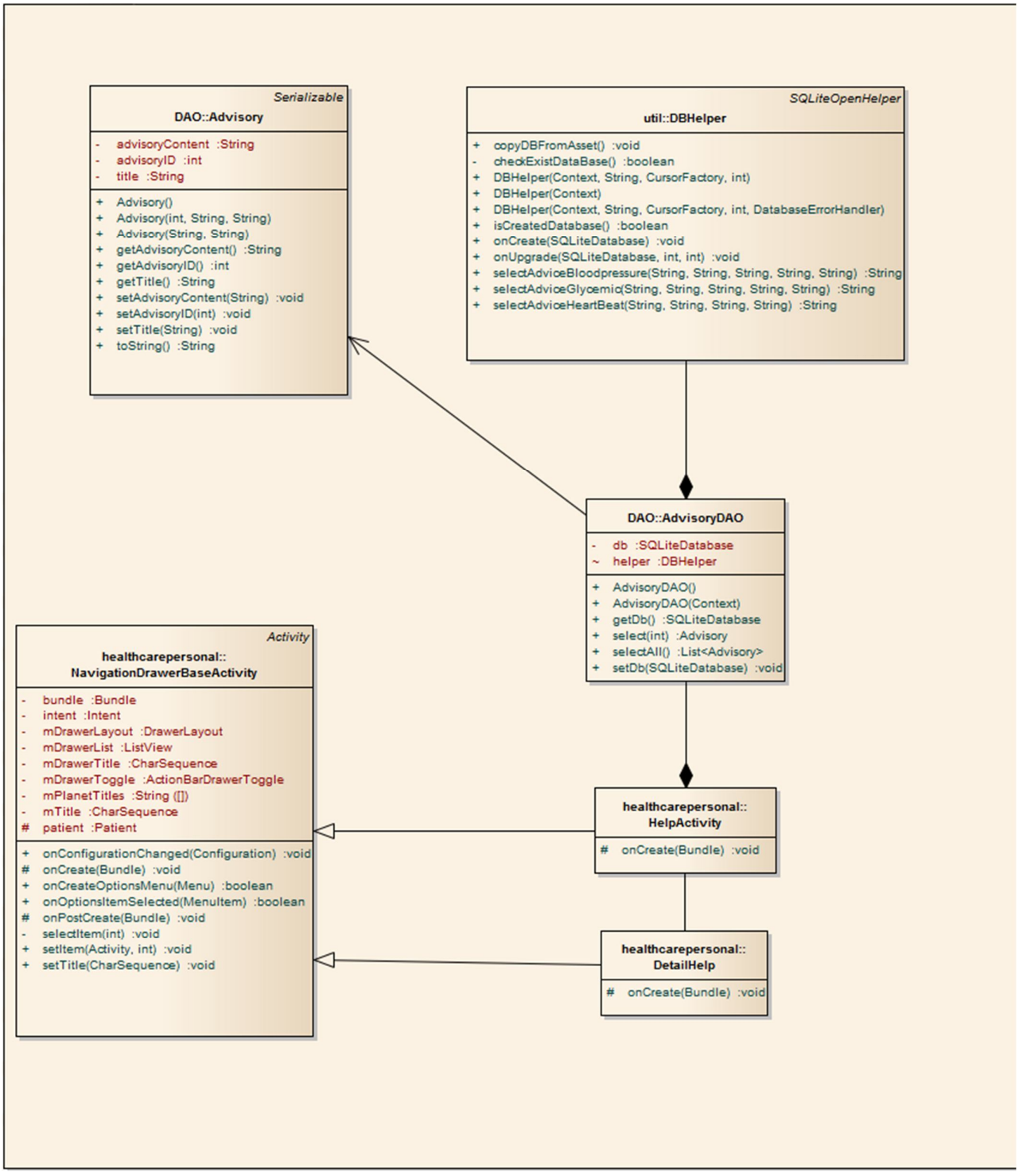



Figure 4-7 Map diagram

##### 4.2.1.5.1 AdvisoryDAO

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| db | SQLiteDatabase | private | The database which this class will work with |
| helper | DBHelper | None | The class helps AdvisoryDAO work with database |

❖ **Methods**

| **Name**: select () | | | | |
|---|---|---|---|---|
| • **Return:** Advisory<br>• **Visibility:** public<br>• **Purpose:** Get and return the Advisory by the Id of that patient | | | | |
| **Parameters and return**: return Advisory which has the Id equal id | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | id | int | In | The Id of the Advisory which need to be gotten |

| **Name**: selectAll () |
|---|
| • **Return:** List< Advisory ><br>• **Visibility:** public<br>• **Purpose:** Get and return the List of all Advisory in the database**:** |
| **Parameters and return**: return the List of all Advisory |

#### 4.2.1.5.2 Advisory

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| advisoryId | int | private | The Id for each Advisory record |
| title | String | private | The title of each Advisory |
| advisoryContent | String | private | The content of the Advisory |

#### 4.2.1.5.3 HelpActivity

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| lvHelper | listView | private | The ListView to display the List of Advisory |
| helper | DBHelper | private | The DBHelper to help the function to work with database |

| | | | |
|---|---|---|---|
| advDAO | AdvisoryDAO | private | The Advisory to help the function to work with database about Advisory |
| adv | List<Advisory> | private | The List of Advisory |
| intentPat | Intent | private | The Intent which need to get the variable pass from the last screen |
| bundlePat | Bundle | private | The bundle which need to get the variable pass from the last screen |

#### 4.2.1.5.4 DetailHelp

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| adv | Advisory | private | The Advisory is displayed in this screen |
| intentAdv | Intent | private | The Intent which need to get the variable pass from the last screen |
| bundle | Bundle | private | The bundle which need to get the variable pass from the last screen |

### *4.2.1.6 Other*

#### 4.2.1.6.1 Date Util

❖ **Methods**

| Name: **DateUtil()** |
|---|
| • **Return:** none<br>• **Visibility:** public<br>• **Purpose:** initialize the instances of DateUtil class |
| **Parameters and return:** None |

| **Name**: nextDay() | | | | |
|---|---|---|---|---|
| • **Return:** Calendar<br>• **Visibility:** public<br>• **Purpose:** initialize the instances of DateUtil class | | | | |
| **Parameters and return**: return the Calendar c which presents the nextday after calculating | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | now | Calendar | In | An instance of Calendar class which will be used to obstain the day when the fuction is used. |

| | | | | |
|---|---|---|---|---|
| 02 | days | Int | In | The number of day which is input from Add Reminder screen for calculating the next day when the reminder is alarmed |

#### 4.2.1.6.2 PatientAdapter

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| listPatient | List<Patient> | Private | List of Patients |
| activity | Actiity | Private | The activity which will use this class |
| inflater | LayoutInflater | Private | The inflater will be need |

❖ **Methods**

**Name**: getCount()

- **Return:** int
- **Visibility:** public
- **Purpose:** Calculate listPatient's number of item

**Parameters and return**: return the size of the listPatient (return 0 if listPatient is null)

**Name**: getItem()

- **Return:** Object
- **Visibility:** public
- **Purpose:** Get and return the object which is at that position

**Parameters and return**: return the Object at the position was chosen

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | position | Int | In | The number stands for the position was chosen. |

**Name**: getItemId()

- **Return:** long
- **Visibility:** public
- **Purpose:** Get and return the number which is at that position

**Parameters and return**: return the number stands for the position was chosen.

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | position | int | In | The number stands for the position was |

| | | | | |
|---|---|---|---|---|
| | | | | chosen. |

**Name**: getListPatient()

- **Return:** List<Patient>
- **Visibility:** public
- **Purpose:** Return the Patient's List

**Parameters and return**: Return the Patient's List

**Name**: setListPatient()

- **Return:** void
- **Visibility:** public
- **Purpose:** Set the information into listPatient

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | listPatient | List<Patient> | In | The List of Patient |

**Name**: PatientAdapter()

- **Return:** none
- **Visibility:** public
- **Purpose:** initialize the instances of PatientAdapter class

**Parameters and return**: none

**Name**: PatientAdapter()

- **Return:** none
- **Visibility:** public
- **Purpose:** initialize the instances of PatientAdapter class with information of listPatient and activity

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | listPatient | List<Patient> | In | The List of Patient |
| 02 | activity | Activity | In | The Activity calls this class |

**Name**: getView()

- **Return:** View
- **Visibility:** public
- **Purpose:** where to set up the informations will be displayed in the List View calls this adapter and how to display them

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | position | Int | In | The number stands for the position was chosen |
| 02 | convertView | View | In | The View call this method |
| 03 | parent | ViewGroup | In | The ViewGroup which is needed |

#### 4.2.1.6.3 ReminderListItemAdapter

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| rliList | List<ReminderListItem> | Private | List of ReminderListItem |
| activity | Actiity | Private | The activity which will use this class |
| inflater | LayoutInflater | Private | The inflater will be need |

❖ **Methods**

**Name**: getCount()

- **Return:** int
- **Visibility:** public
- **Purpose:** Calculate List of ReminderListItem's number of item

**Parameters and return**: return the size of the rliList (return 0 if rliList is null)

**Name**: getItem()

- **Return:** Object
- **Visibility:** public
- **Purpose:** Get and return the object which is at that position

**Parameters and return**: return the Object at the position was chosen

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | position | int | In | The number stands for the position was chosen. |

**Name**: getItemId()

- **Return:** long
- **Visibility:** public
- **Purpose:** Get and return the number which is at that position

**Parameters and return**: return the number stands for the position was chosen.

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | position | int | In | The number stands for the position was chosen. |

**Name**: getRliList()

- **Return:** List<ReminderListItem>
- **Visibility:** public
- **Purpose:** Return the ReminderListItem's List

**Parameters and return**: Return the ReminderListItem's List

**Name**: setRlilist()

- **Return:** void
- **Visibility:** public
- **Purpose:** Set the information into ReminderListItem's List

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | listPatient | List< ReminderListItem > | In | The List of ReminderListItem |

**Name**: ReminderListItemAdapter()

- **Return:** none
- **Visibility:** public
- **Purpose:** initialize the instances of ReminderListItemAdapter class

**Parameters and return**: none

**Name**: ReminderListItemAdapter()

- **Return:** none
- **Visibility:** public
- **Purpose:** initialize the instances of ReminderListItemAdapter class with information of rliList and

| activity | | | | |
|---|---|---|---|---|
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | rliList | List<ReminderListItem> | In | The List of ReminderListItem |
| 02 | activity | Activity | In | The Activity calls this class |

| **Name**: getView() | | | | |
|---|---|---|---|---|
| • **Return:** View<br>• **Visibility:** public<br>• **Purpose:** where to set up the informations will be displayed in the List View calls this adapter and how to display them | | | | |
| **Parameters and return**: The View to display customized view | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | position | Int | In | The number stands for the position was chosen |
| 02 | convertView | View | In | The View call this method |
| 03 | parent | ViewGroup | In | The ViewGroup which is needed |

#### 4.2.1.6.4 ReminderListItemAdapter : ViewHolder

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| name | TextView | private | The Textview which will display the information of the Medicine's Name |
| quantity | TextView | private | The Textview which will display the information of the Medicine's Quantity |

#### 4.2.1.6.5 RecenltyStatusAdapter

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| rsList | List<RecentlyStatus> | Private | List of RecentlyStatus |
| activity | Actiity | Private | The activity which will use this class |
| inflater | LayoutInflater | Private | The inflater will be need |

❖ **Methods**

**Name**: getCount()

- **Return:** int
- **Visibility:** public
- **Purpose:** Calculate List of RecentlyStatus's number of item

**Parameters and return**: return the size of the rsList (return 0 if rsList is null)

**Name**: getItem()

- **Return:** Object
- **Visibility:** public
- **Purpose:** Get and return the object which is at that position

**Parameters and return**: return the Object at the position was chosen

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | position | int | In | The number stands for the position was chosen. |

**Name**: getItemId()

- **Return:** long
- **Visibility:** public
- **Purpose:** Get and return the number which is at that position

**Parameters and return**: return the number stands for the position was chosen.

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | position | int | In | The number stands for the position was chosen. |

**Name**: getRliList()

- **Return:** List< RecentlyStatus >
- **Visibility:** public
- **Purpose:** Return the RecentlyStatus 's List

**Parameters and return**: Return the RecentlyStatus 's List

**Name**: setRslist()

- **Return:** void
- **Visibility:** public
- **Purpose:** Set the information into RecentlyStatus List

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | listPatient | List< RecentlyStatus > | In | The List of RecentlyStatus |

**Name**: RecentlyStatusAdapter()

- **Return:** none
- **Visibility:** public
- **Purpose:** initialize the instances of RecentlyStatusAdapter class

**Parameters and return**: none

**Name**: RecentlyStatusAdapter ()

- **Return:** none
- **Visibility:** public
- **Purpose:** initialize the instances of RecentlyStatusAdapter class with information of rsList and activity

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | rsList | List<RecentlyStatus> | In | The List of RecentlyStatus. |
| 02 | activity | Activity | In | The Activity calls this class |

**Name**: getView()

- **Return:** View
- **Visibility:** public
- **Purpose:** where to set up the informations will be displayed in the List View calls this adapter and how to display them

**Parameters and return**: The View to display customized view

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | position | Int | In | The number stands for the position was chosen |
| 02 | convertView | View | In | The View call this method |
| 03 | parent | ViewGroup | In | The ViewGroup which is needed |

#### 4.2.1.6.6 DBUtil

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| context | Context | private | The context which is needed |

❖ **Methods**

| **Name**: copyDBFromAsset () |
|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** copy database from asset into project |
| **Parameters and return**: none |

#### 4.2.1.6.7 TimeAdapter

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| act | Activiy | private | The activity which will use this class |
| inf | LayoutInflater | private | The inflater will be need |
| listTime | List<String> | private | The List of Time |

❖ **Methods**

| **Name**: add () | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** add an element in to List Of Time | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | s | String | In | The String contain the time to add to list |

#### 4.2.1.6.8 TimeAdapter : Holder

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| img | ImageButton | private | The Image Button Delete |
| time | TextView | private | The TextView to display Time |

#### 4.2.1.6.9 AlarmDAO

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| db | SQLiteDatabase | private | The database which this class will work with |
| helper | DBHelper | None | The class helps HospitalDAO work with database |

❖ **Methods**

**Name**: selectAlarmItems ()

- **Return:** List<AlarmItem>
- **Visibility:** public
- **Purpose:** Get and return the List of AlarmItem which will be alarmed today

**Parameters and return**: return the List of AlarmItem

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | today | String | In | The String which present the day this method is called |

**Name**: selectAllByReminderID ()

- **Return:** List< Alarm >
- **Visibility:** public
- **Purpose:** Get and return the List of all Alarm belongs to the Reminder which has ID equal reminderID

**Parameters and return**: return the List of all Alarm

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | reminderID | int | In | The number present the ID of the reminder which the alarm belongs to |

**Name**: delete ()

- **Return:** void
- **Visibility:** public
- **Purpose:** Remove the an Alarm from database

**Parameters and return**:

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 01 | alarm | Alarm | In | The Alarm need to be removed from database |

| **Name**: delete () | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Remove the an Alarm from database | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | alarmID | int | In | The Id of the Alarm need to be removed from database |

| **Name**: insert () | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Insert the information of inputted Alarm into database | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | alarm | Alarm | In | The Alarm to insert into database |

| **Name**: update () | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Update the information of an exist Alarm with inputted information | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | alarm | Alarm | In | The Alarm contain the information to be update to the record of Alarm which has the same ID |

#### 4.2.1.6.10 MedicineDAO

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| db | SQLiteDatabase | private | The database which this class will work with |
| helper | DBHelper | None | The class helps HospitalDAO work with database |

❖ **Methods**

| **Name**: insert() | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Insert the information of inputted Medicine into database | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | medicine | Medicine | In | The Medicine to insert into database |

| **Name**: selectMedicineName () | | | | |
|---|---|---|---|---|
| • **Return:** String<br>• **Visibility:** public<br>• **Purpose:** Get and return the Name of the Medicine which belongs to a Reminder with reminderID | | | | |
| **Parameters and return**: return Medicine's Name | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | reminderID | int | In | The Id of the Reminder which the medicine belongs to |

| **Name**: update () | | | | |
|---|---|---|---|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Update the information of an exist Medicine with inputted information | | | | |
| **Parameters and return**: | | | | |
| **No** | **Parameter** | **Type** | **In/Out** | **Description** |
| 01 | medicine | Medicine | In | The Medicine contain the information to be update to the record of Medicine which has the same ID |

| **Name**: select () |
|---|

- **Return:** Medicine
- **Visibility:** public
- **Purpose:** Get and return the Medicine by the Id of that Medicine

**Parameters and return**: return Medicine which has the Id equal id

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | id | int | In | The Id of the Medicine which need to be gotten |

**Name**: selectMedicineID ()

- **Return:** int
- **Visibility:** public
- **Purpose:** Get and return the Id of the Medicine which belongs to a Reminder with reminderID

**Parameters and return**: return Medicine's Id

| No | Parameter | Type | In/Out | Description |
|---|---|---|---|---|
| 01 | reminderID | int | In | The Id of the Reminder which the medicine belongs to |

**Name**: selectAll ()

- **Return:** List< Medicine >
- **Visibility:** public
- **Purpose:** Get and return the List of all Medicine in the database

**Parameters and return**: return the List of all Medicine

**Name**: selectLastID ()

- **Return:** int
- **Visibility:** public
- **Purpose:** Get and return the ID of the last record of Medicine in database

**Parameters and return**: return the ID of the last record of Medicine

#### 4.2.1.6.11 Alarm

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| alarmID | int | private | The Id for each Alarm record in database |

| | | | |
|---|---|---|---|
| reminderID | int | private | The Id of the Reminder which the Alarm belongs to |
| medicineID | int | private | The Id of the medicine in the Reminder which the Alarm belongs to |
| time | String | private | The time to alarm |

#### 4.2.1.6.12 Medicine

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| medicineID | int | private | The Id for each Medicine record in database |
| name | String | private | The name of the Medicine |

#### 4.2.1.6.13 RecentlyStatus

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| recentlyStatusID | int | private | The Id for each RecentlyStatus record in database |
| patientID | int | private | The Id of the Patient who the RecentlyStatus belongs to |
| glycemic | String | private | The index of the Glycemic is chosen |
| status | int | private | The index of the Status is chosen (0 is "Trước ăn", 1 í "Sau ăn") |
| heartBeat | String | private | The index of the HeartBeat is chosen |
| maxBloodPressure | String | private | The index of the MaxBloodPressure is chosen |
| minBloodPressure | String | private | The index of the MinBloodPressure is chosen |
| date | String | private | The date when the recently status was recorded |
| time | String | private | The time when the recently status was recorded |
| advice | String | private | The advice was given base on chosen index |

#### 4.2.1.6.14 BinarySelection

- **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| binary | int | private | The number present for the selection |
| selection | String | private | The String contain the content of the selection |

#### 4.2.1.6.15 ReminderListItem

- **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| medicineID | int | private | The Id of the Medicine |
| medicineName | String | private | The name of the Medicine |
| total | int | private | The quantity of the Medicine |

#### 4.2.1.6.16 AlarmItem

- **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| patientName | String | private | The name of the Patient who this AlarmItem belogs to |
| medicineName | String | private | The Name of the Medicine |
| dosage | int | private | The dosage of the Medicine |
| note | String | private | The note for the reminder |
| time | string | private | The time to alarm the reminder |

#### 4.2.1.6.17 TypeDAO

- **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| db | SQLiteDatabase | private | The database which this class will work with |
| helper | DBHelper | None | The class helps HospitalDAO work with database |

- **Methods**

| **Name**: selectAll() |
|---|
| • **Return:** List< Type > |

| |
|---|
| • **Visibility:** public<br>• **Purpose:** Get and return the List of all Type in the database |
| **Parameters and return**: |

#### 4.2.1.6.18 Type

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| typeId | int | private | The Id of the Type in datase |
| lowerAge | int | private | The lower bound for a tupe of Age |
| upperAge | int | private | The upper bound for a tupe of Age |
| personType | int | private | The Type of a person (0 is "Bình thường", 1 is "Hay vận động") |
| personType1 | String | private | The Type of a person ("Bình thường"/"Hay vận động") |
| gender | int | private | The gender for each Type record (0 is "Nam", 1 is "Nữ") |

#### 4.2.1.6.19 MenuActivity

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| adapter | PatientAdapter | private | The Adapter to help display Patient List |
| btnJudgement | Button | private | The Button to pass to the Judgement screen |
| btnListReminder | Button | private | The Button to pass to the List Reminder screen |
| btnShowHospital | Button | private | The Button to pass to Hospital screen |
| bundle | Bundle | private | The bundle which need to get the variable pass from the last screen |
| intent | Intent | private | The Intent which need to get the variable pass from the last screen |
| patient | Patient | private | The patient's instance to get the patient pass from the last screen |
| helper | DBHelper | private | The DBHelper to help woking with database |
| imgHospital | ImageView | private | The ImageView to display the image of Hospital function |
| imgRem | ImageView | private | The ImageView to display the image of |

| | | | |
|---|---|---|---|
| | | | Reminder function |
| imgJudge | ImageView | private | The ImageView to display the image of Judgement function |
| list | List<Patient> | private | The List of Patient |
| listView | ListView | private | The list view to display Patient List |
| patientDAO | PatientDAO | private | The PatientDAO to work with database about Patient |

❖ **Methods**

| **Name**: loadDataFromAsset () |
|---|
| • **Return:** void<br>• **Visibility:** public<br>• **Purpose:** Access in to asset and load the images to set on ImageViews |
| **Parameters and return**: none |

#### 4.2.1.6.20 AboutActivity

❖ **Attributes**

| Attributes | Type | Visibility | Description |
|---|---|---|---|
| about | TextView | private | The TextView to display the information About us |
| scroll | ScrollView | private | The ScrollView to TextView to be able to be scroll up and down when the content it out of the screen |

## 4.2.2 Process view

### 4.2.2.1 First time login

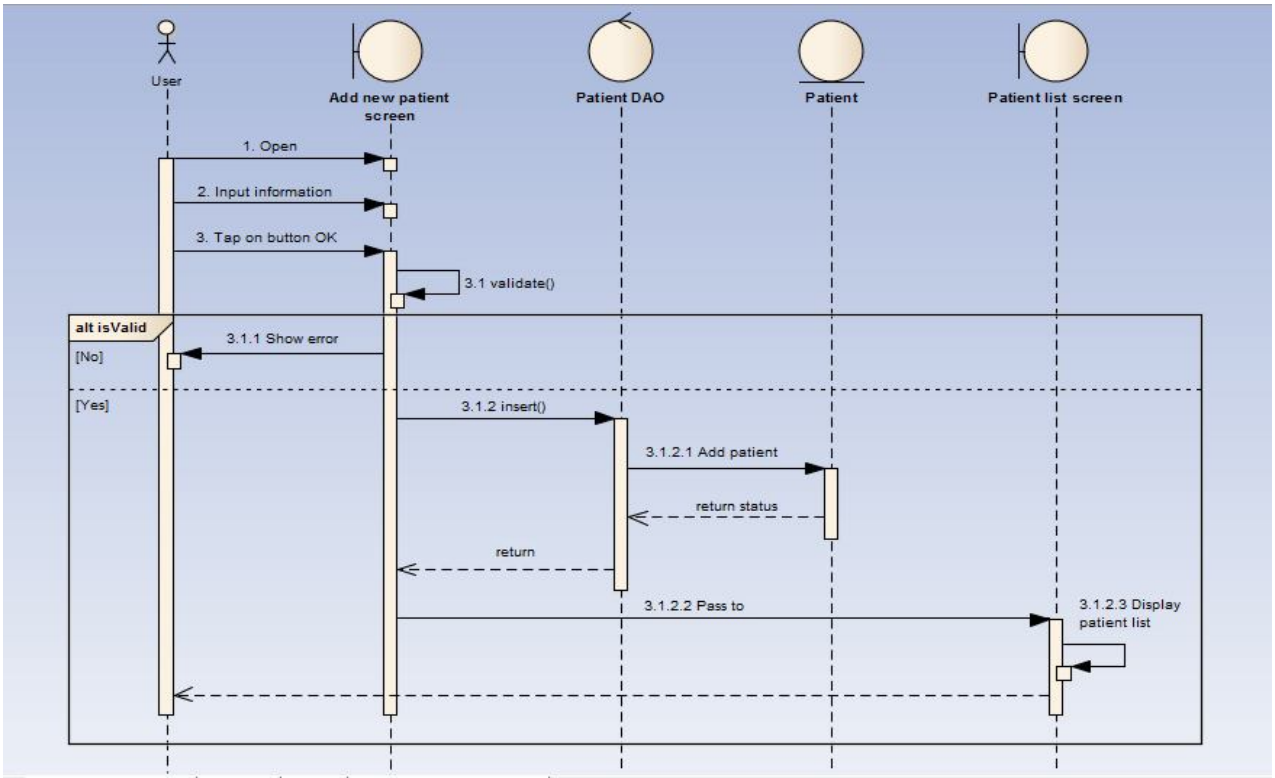



Figure 4-8 First time login sequence diagram

### 4.2.2.2 Patient

#### 4.2.2.2.1 Patient list

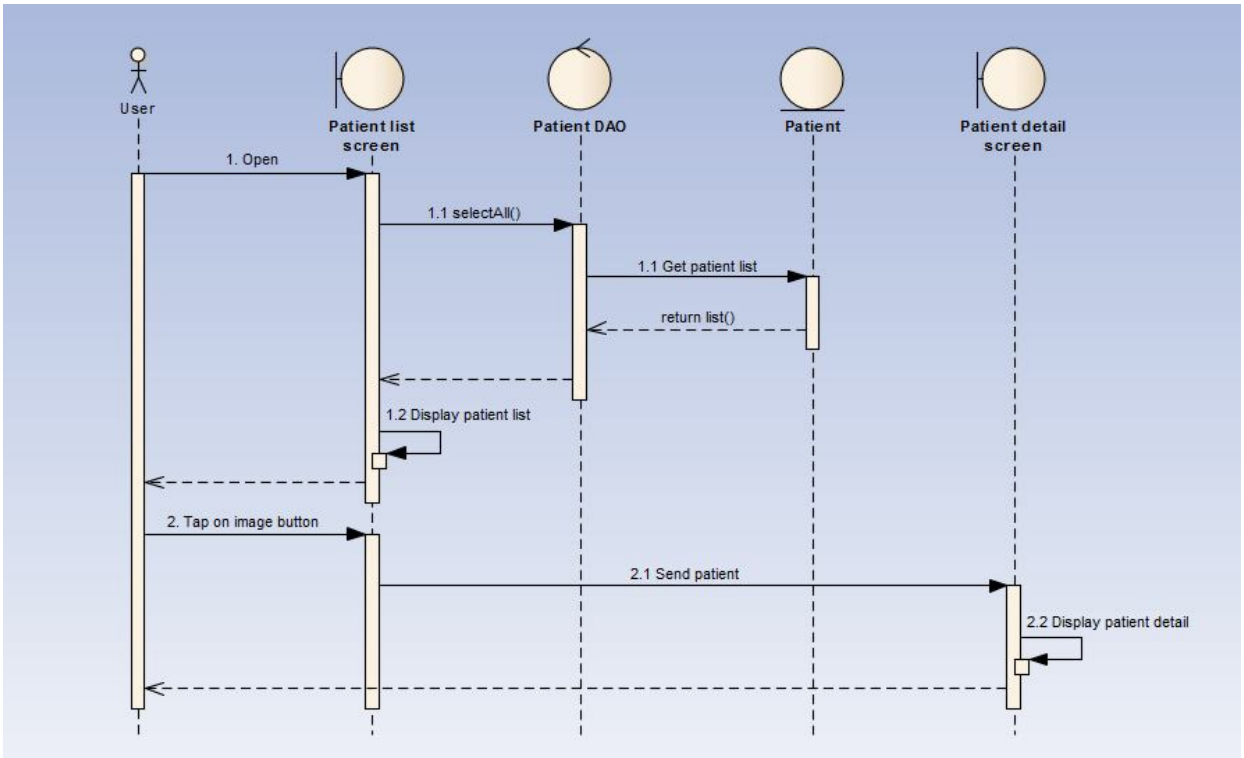



Figure 4-9 View patient list sequence diagram

#### 4.2.2.2.2 Add new patient

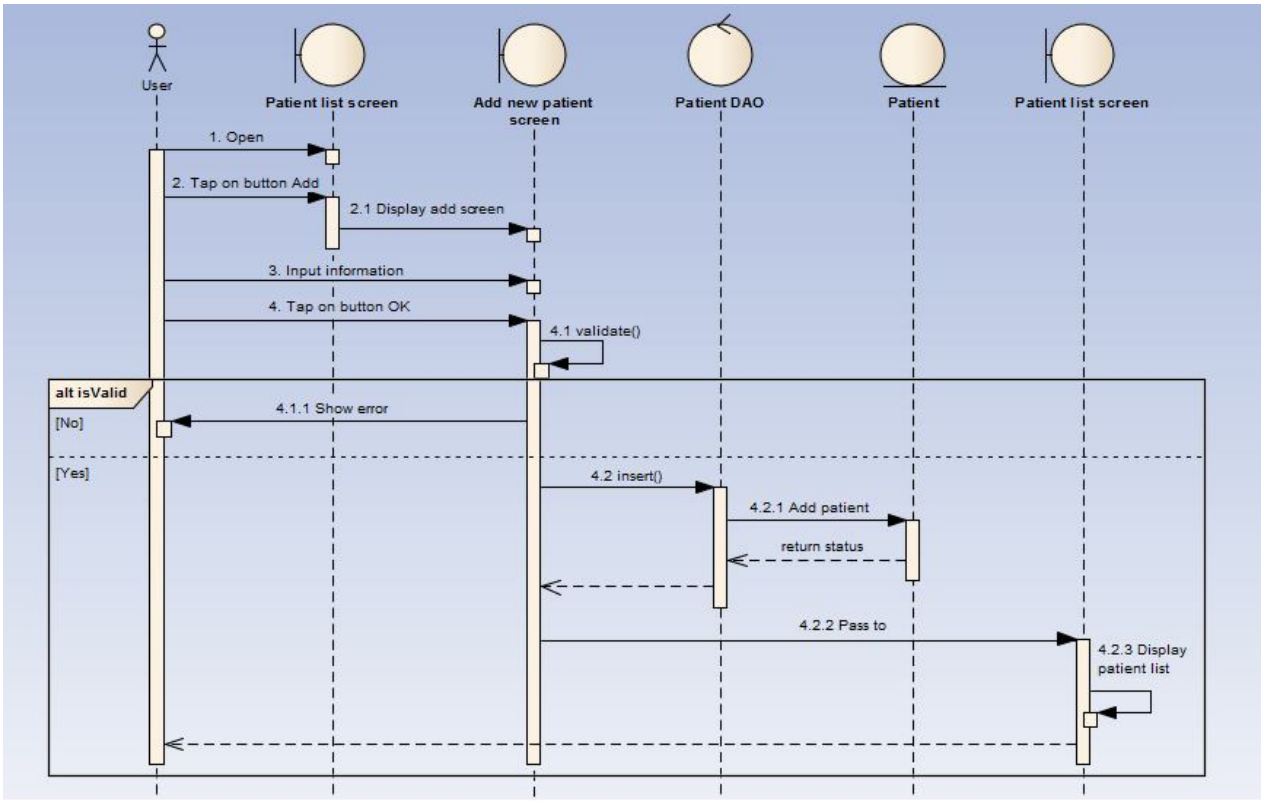



Figure 4-10 Add new patient sequence diagram

#### 4.2.2.2.3 Edit patient

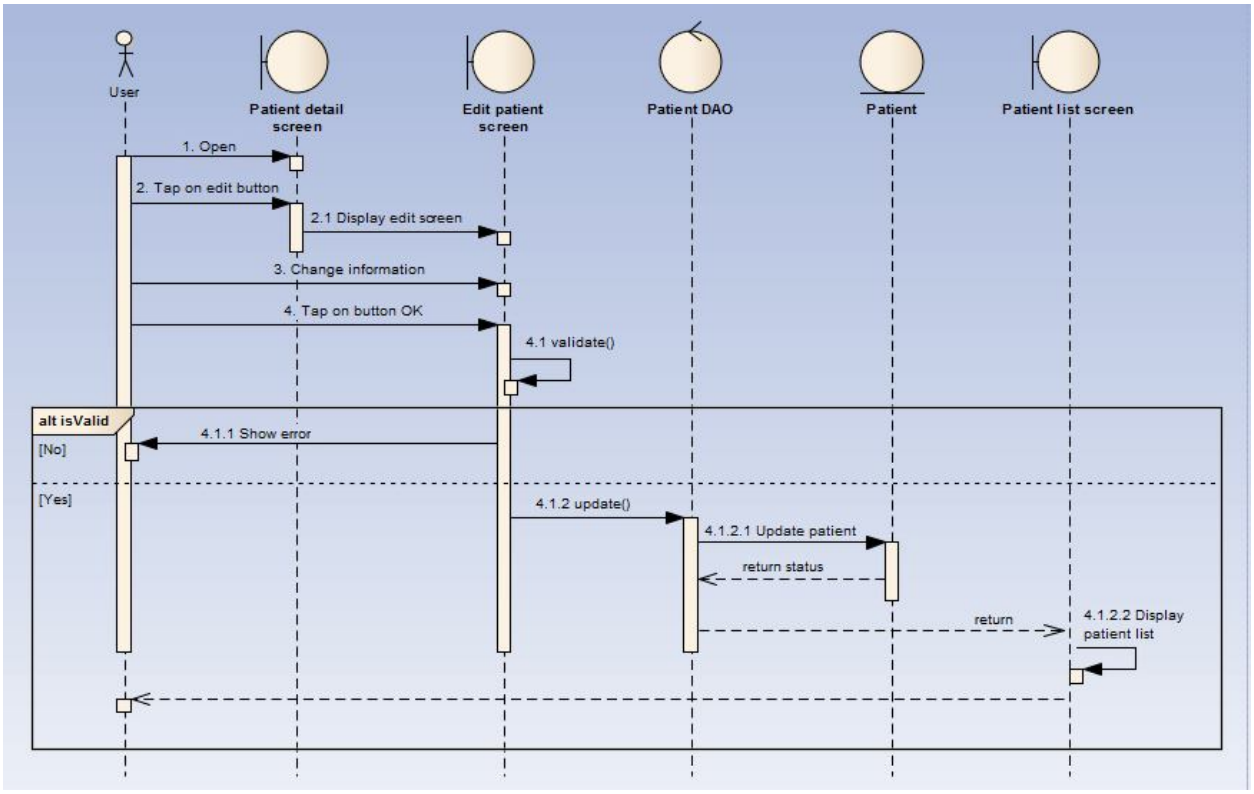



Figure 4-11 Edit patient sequence diagram

#### 4.2.2.2.4 Delete patient

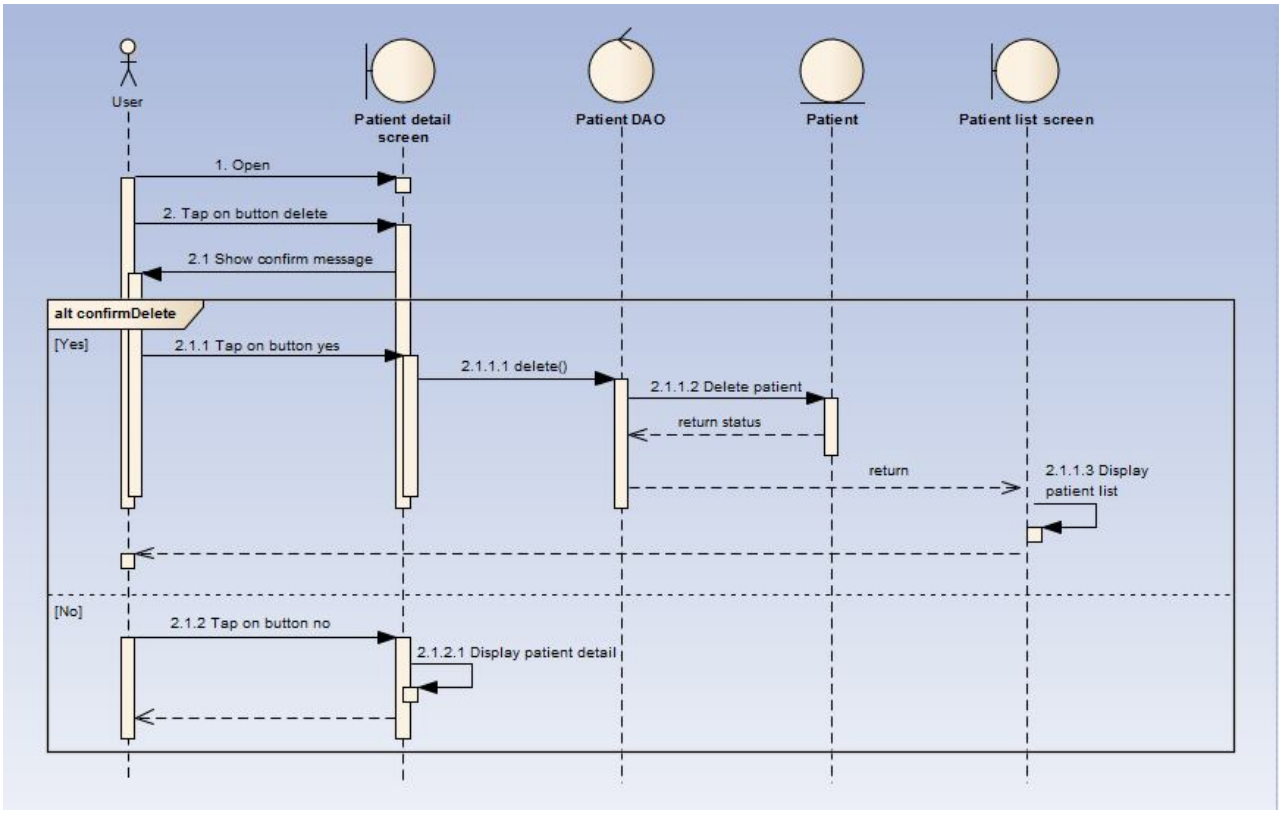



Figure 4-12 Delete patient sequence diagram

#### 4.2.2.3 Health judgment

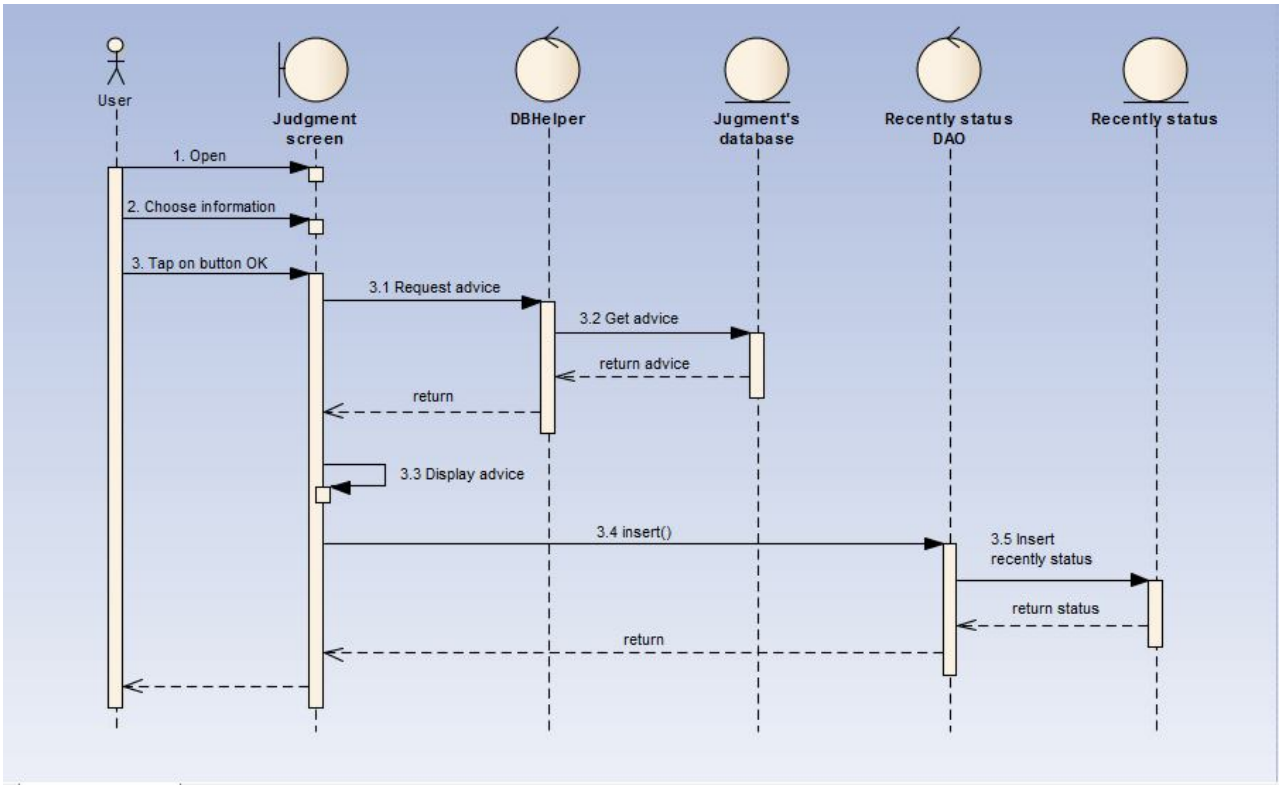



Figure 4-13 Health judgment sequence diagram

#### 4.2.2.4 Recently status

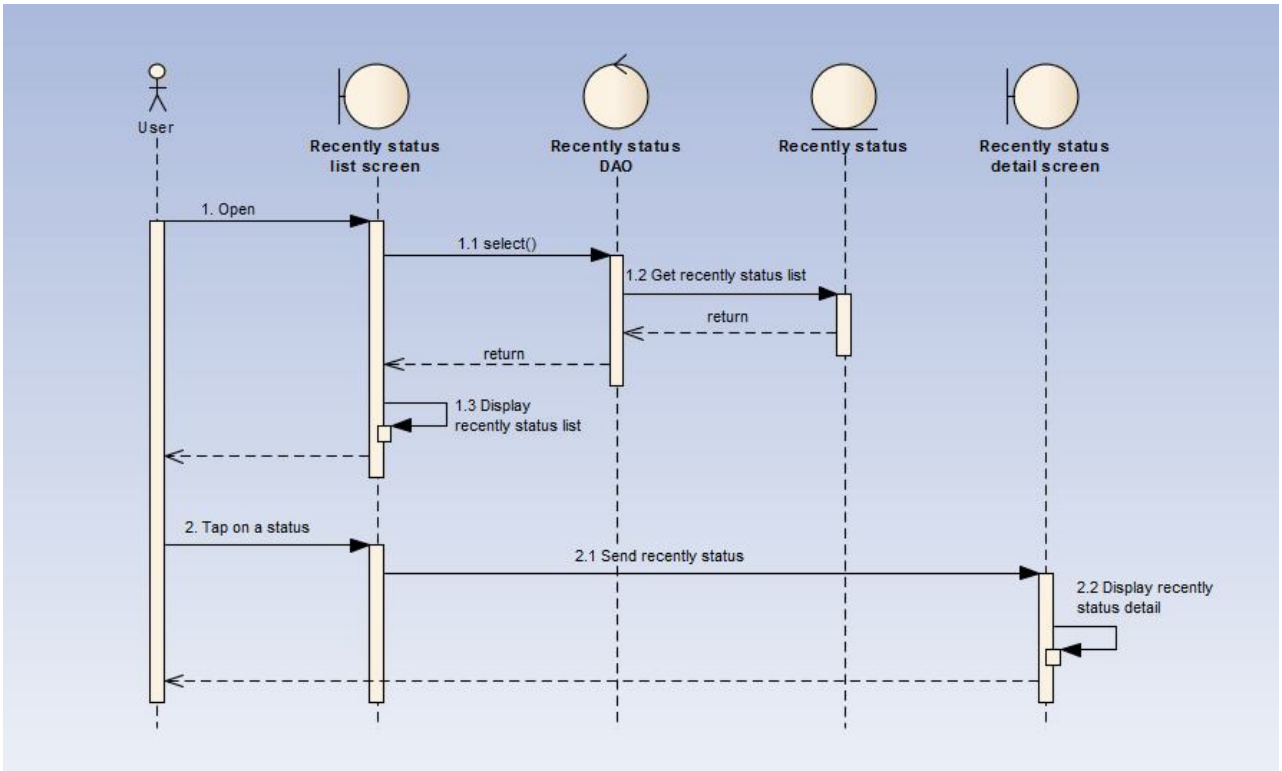



Figure 4-14 Recently status sequence diagram

### 4.2.2.5 Reminder

#### 4.2.2.5.1 Reminder list

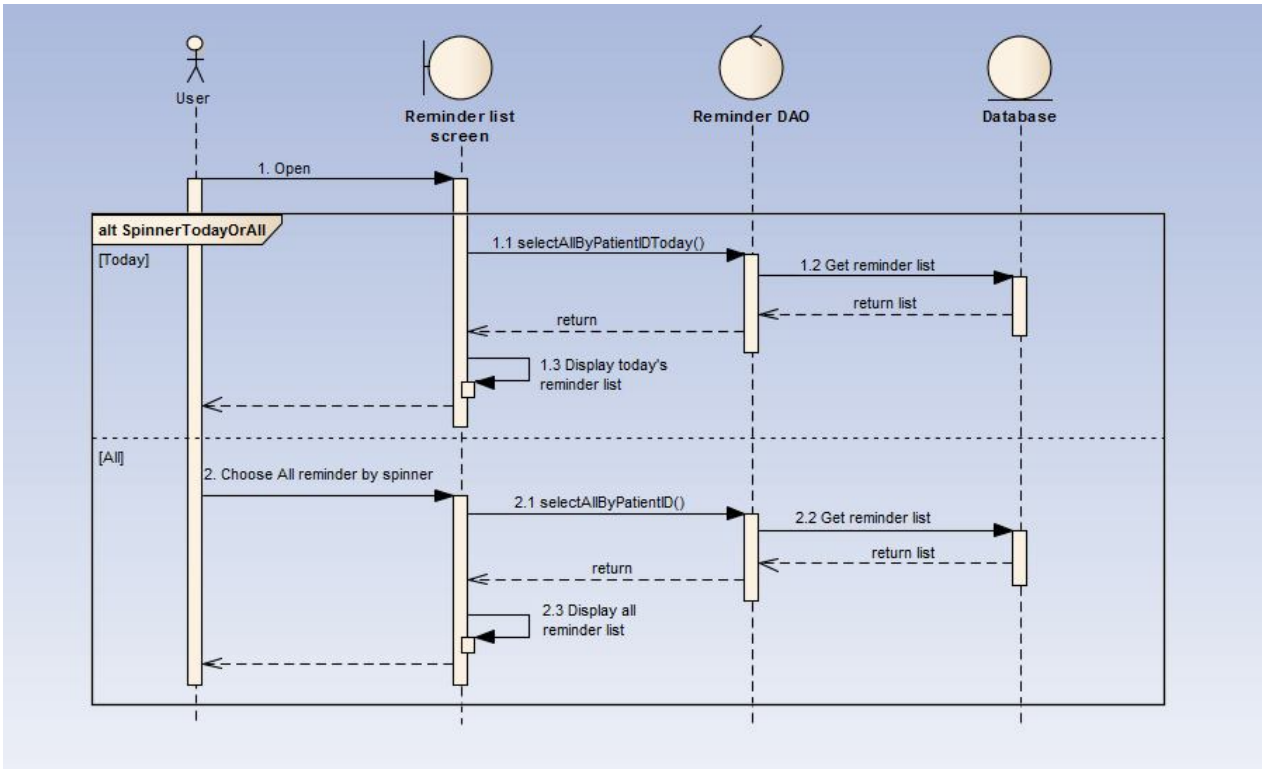



Figure 4-15 View reminder list sequence diagram

#### 4.2.2.5.2 Add new reminder

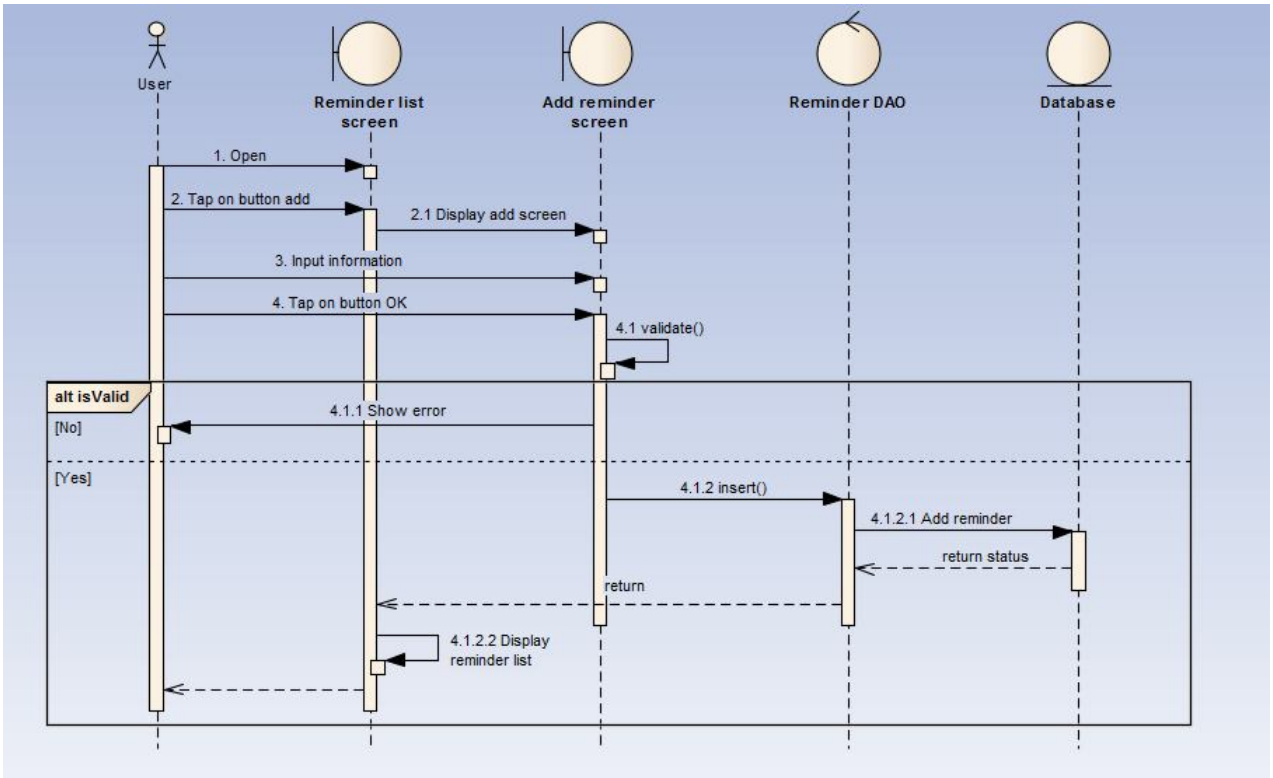



Figure 4-16 Add new reminder sequence diagram

#### 4.2.2.5.3 Edit reminder

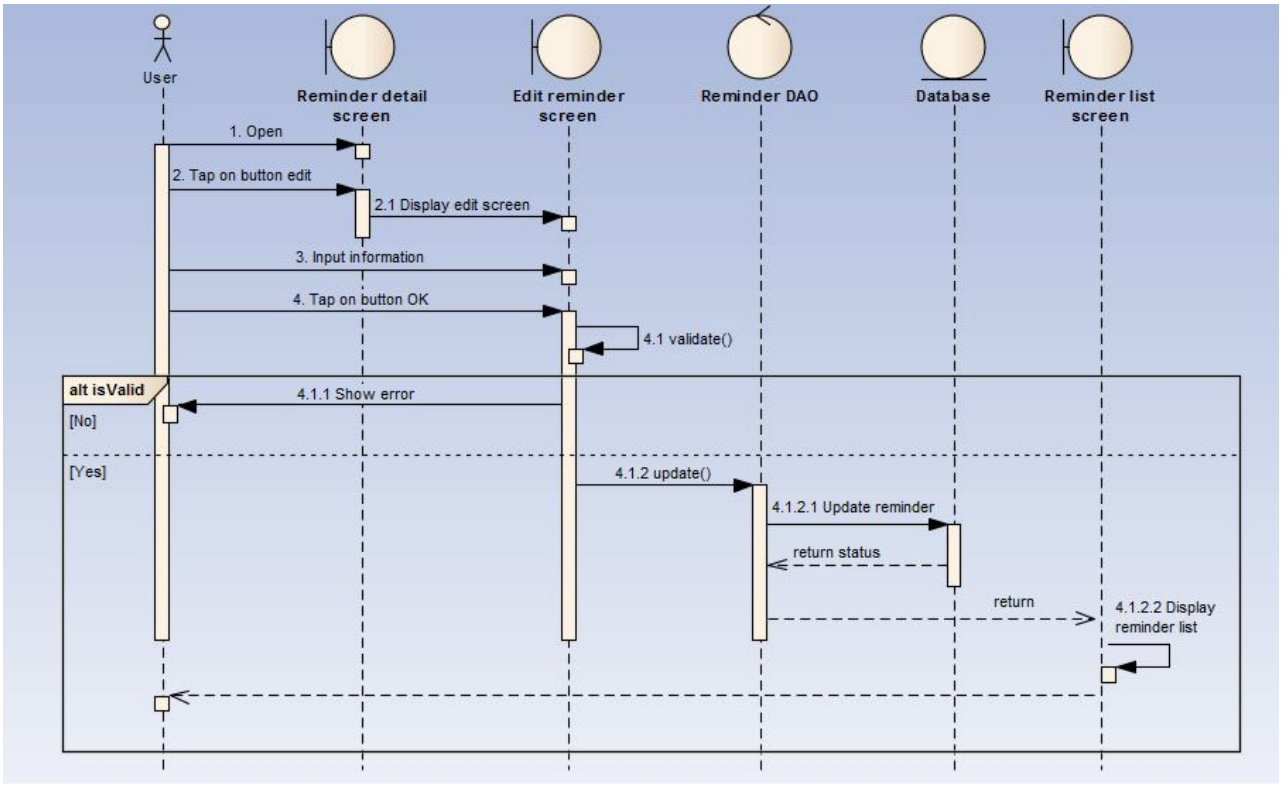



Figure 4-17 Edit reminder sequence diagram

#### 4.2.2.5.4 Delete reminder

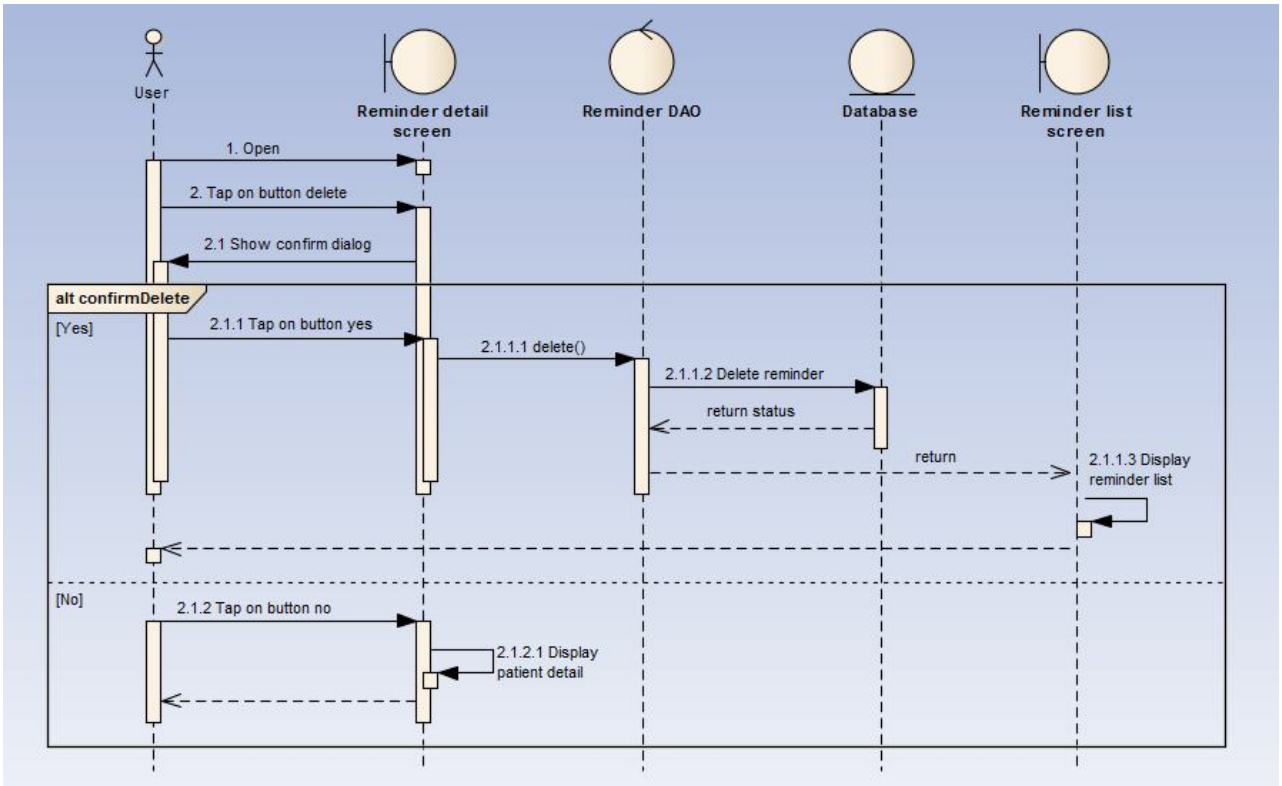



Figure 4-18 Delete reminder sequence diagram

#### 4.2.2.6 Hospital

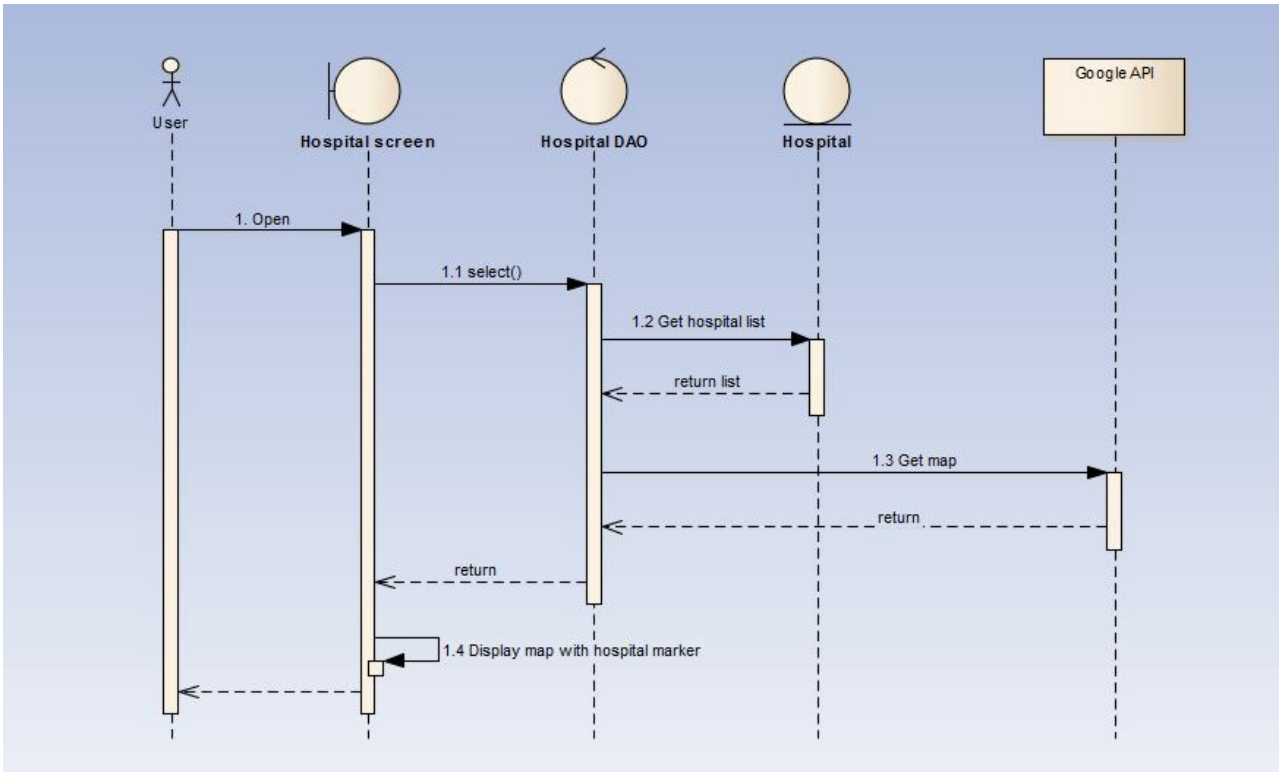



Figure 4-19 Hospital sequence diagram

#### 4.2.2.7 Help

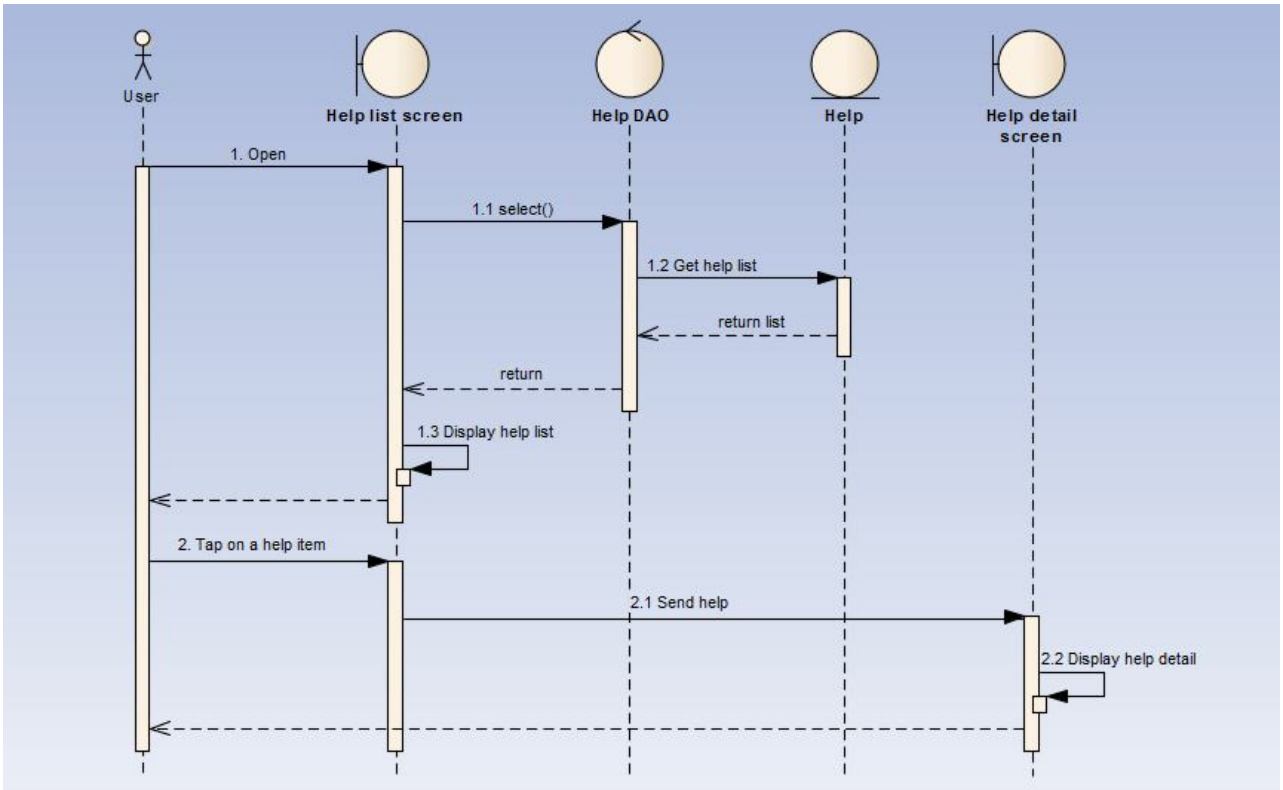



Figure 4-20 Help sequence diagram

## 4.3 Screen Design

### 4.3.1 Screen flow

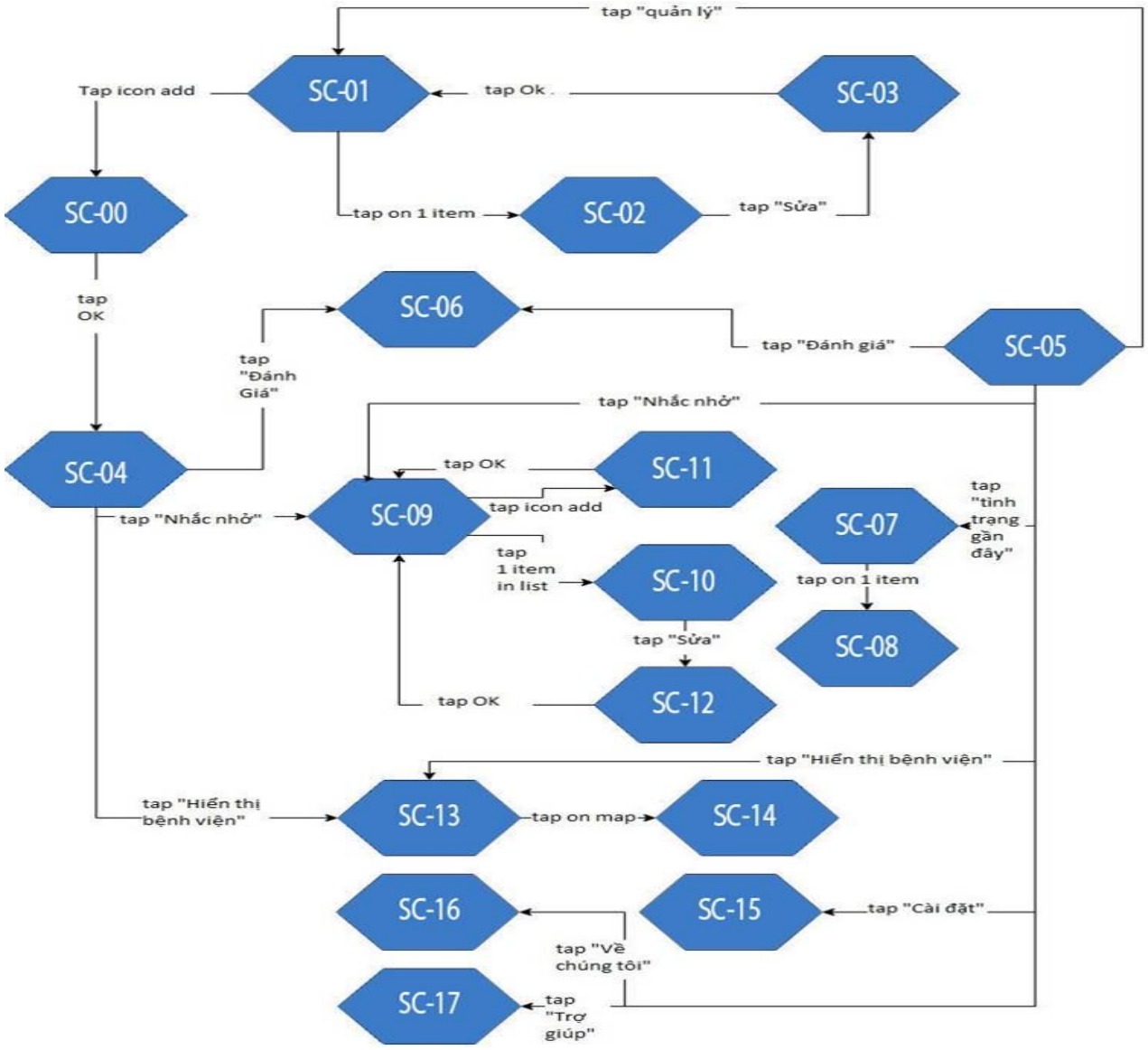



Figure 4-21 Screen flow

TABLE 4-1 Screen list

| No. | Screen name | Screen no. | Associate Use-cases |
|---|---|---|---|
| 1 | “Thêm bệnh nhân” Screen | SC-00 | UC-01, UC-03 |
| 2 | “Danh sách bệnh nhân” Screen | SC-01 | UC-02 |
| 3 | “Chi tiết bệnh nhân” Screen | SC-02 | UC-05 |
| 4 | “Chỉnh sửa thông tin” Screen | SC-03 | UC-06 |
| 5 | Main Screen | SC-04 | UC-08, UC-10, UC-15, UC-02 |
| 6 | Navigator drawer | SC-05 | UC-02, UC-08, UC-09, UC-10, UC-15, UC-16, UC-17, UC-18 |
| 7 | “Đánh giá” Screen | SC-06 | UC-08, UC-02 |
| 8 | “Tình trạng gần đây” Screen | SC-07 | UC-09, UC-02 |
| 9 | “Chi tiết tình trạng” Screen | SC-08 | |
| 10 | “Danh sách nhắc nhở” Screen | SC-09 | UC-10, UC-02 |
| 11 | “Chi tiết lời nhắc nhở” Screen | SC-10 | UC-12, UC-14, UC-02 |
| 12 | “Thêm mới lời nhắc nhở” Screen | SC-11 | UC-11 |
| 13 | “Thay đổi lời nhắc nhở” Screen | SC-12 | UC-13 |
| 14 | “Hiển thị bệnh viện” Screen | SC-13 | UC-15 |
| 15 | Open Map Screen | SC-14 | UC-16 |
| 16 | “Trợ giúp” Screen | SC-16 | UC-17 |
| 17 | “Về chúng tôi” Screen | SC-17 | UC-18 |

### 4.3.2 Screen prototype

#### *4.3.2.1 Main screen*

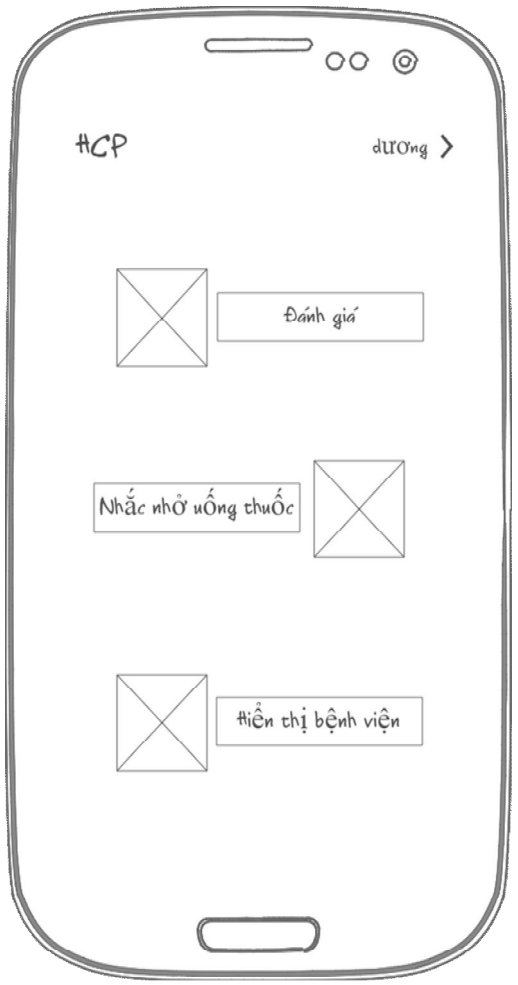



Figure 4-22 Main screen prototype

TABLE 4-2 Main screen description

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | **Action bar** | | | | |
| **1** | Navigation | O | Swipe | - | Navigation drawer is displayed | **-** | |
| **2** | Screen name | O | Label | - | Screen name is displayed | **-** | |
| **3** | dương > | O | Button | - | Current patient is displayed | **-** | |
| | | | **Main content** | | | | |
| **1** | "Đánh giá" | O | Button | - | Judgment screen is displayed | - | |
| **2** | "Nhắc nhở uống thuốc" | O | Button | - | Reminder screen is displayed | - | |
| **3** | "Hiển thị bệnh viện" | O | Button | - | Hospital screen is displayed | - | |

#### 4.3.2.2 Navigation drawer

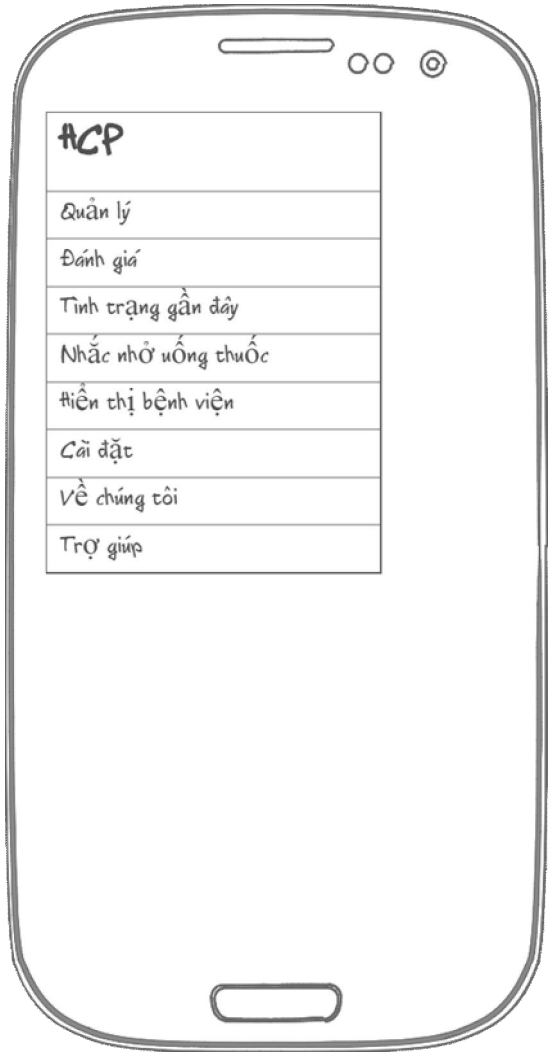



Figure 4-23 Navigation drawer prototype

**TABLE 4-3 Navigation drawer description**

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| **Action bar** | | | | | | | |
| **1** | Refer Main screen | | | | | | |
| **Main content** | | | | | | | |
| 1 | “Quản lý” | O | Button | - | Patient list screen is displayed | - | |
| 2 | “Đánh giá” | O | Button | - | Judgment screen is displayed | - | |
| 3 | “Tình trạng gần đây” | O | Button | - | Recently status screen is displayed | - | |
| 4 | “Nhắc nhở uống thuốc” | O | Button | - | Reminder screen is displayed | - | |
| 5 | “Hiển thị bệnh viện ” | O | Button | - | Hospital screen is displayed | - | |
| 6 | “Cài đặt” | O | Button | - | Setting screen is displayed | - | |
| 7 | “Về chúng tôi” | O | Button | - | About us screen is displayed | - | |
| 8 | “Trợ giúp” | O | Button | - | Help screen is displayed | - | |

#### 4.3.2.3 Patient

##### 4.3.2.3.1 Patient list

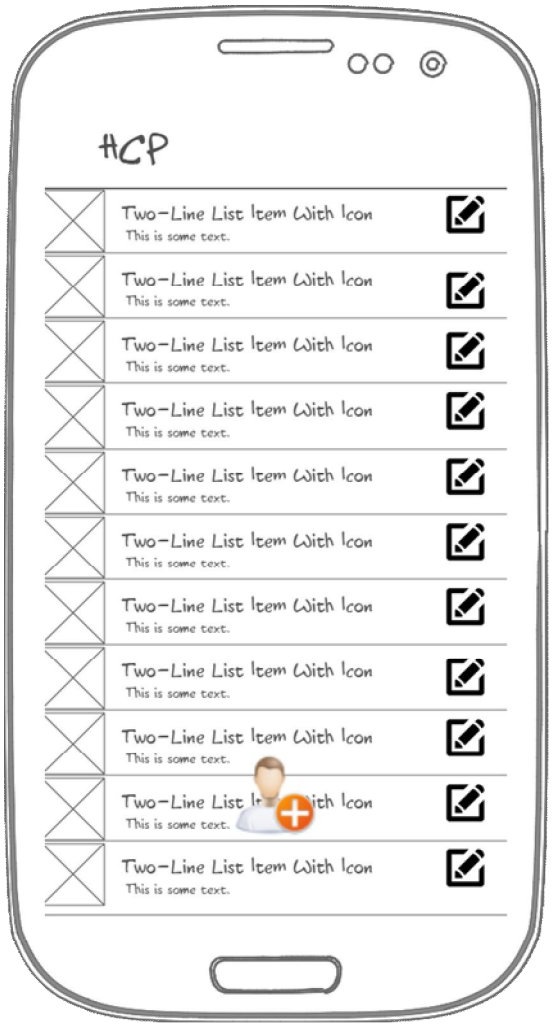



Figure 4-24 Patient list prototype

TABLE 4-4 Patient list description

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | **Action bar** | | | | |
| **1** | - | | | | | | |
| | | | **Main content** | | | | |
| 1 | Patient name | O | List view | - | Name of all patient is displayed | - | |
| 2 | Add button | O | Button | - | Add new patient screen is displayed | - | |
| 3 | | O | Button | - | Patient details screen is displayed | - | |

#### 4.3.2.3.2 Add new patient

Figure 4-25 Add new patient prototype

**TABLE 4-5 Add new patient description**

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | **Action bar** | | | | |
| **1** | - | | | | | | |
| | | | **Main content** | | | | |
| **1** | Name | I | Text field | 30 | User enter patient name | Not have special character or number (@!~#$%^&*?<>:;) | |
| **2** | Year of birth | I | Text field | - | User choose year of birth by number picker | - | Input number only |
| **3** | Gender | I | Radio button | - | User choose gender of patient | - | |
| **4** | Type | I | Spinner | - | User choose type of patient | - | |
| **5** | Reset | I/O | Button | - | Reset all filled fields to default | - | |
| **6** | OK | I/O | Button | - | Save patient information into database | - | |

#### 4.3.2.3.3 Patient detail

Figure 4-26 Patient detail prototype

TABLE 4-6 Patient details description

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| **Action bar** | | | | | | | |
| **1** | - | | | | | | |
| **Main content** | | | | | | | |
| **1** | Name | O | Label | - | Patient name is displayed | - | |
| **2** | Year of birth | O | Label | - | Patient's year of birth is displayed | - | |
| **3** | Gender | O | Label | - | Gender of patient is displayed | - | |
| **4** | Type | O | Label | - | Type of patient is displayed | - | |
| **5** | Delete | O | Button | - | Delete this patient | - | |
| **6** | Edit | O | Button | - | Change information of this patient | - | |

#### 4.3.2.3.4 Edit patient information

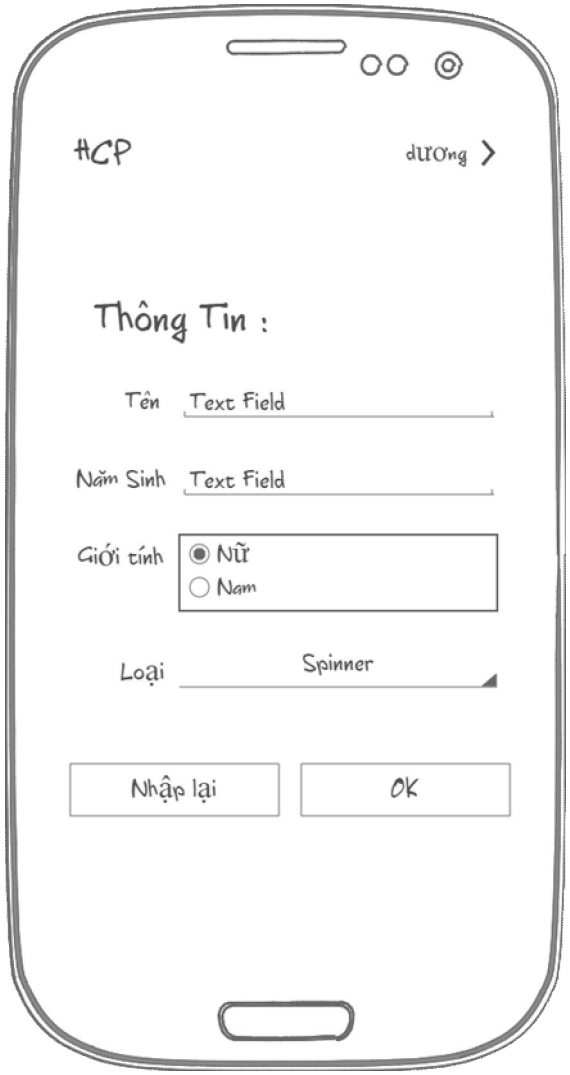



Figure 4-27 Edit patient information prototype

TABLE 4-7 Edit patient information description

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | **Action bar** | | | | |
| **1** | - | | | | | | |
| | | | **Main content** | | | | |
| **1** | Name | I | Text field | 30 | User enter patient name | Not have special character or number (@!~#$%^&*?<>:;) | |
| **2** | Year of birth | I | Date picker | - | User choose year of birth by date picker | - | Input number only |
| **3** | Gender | I | Radio button | - | User choose gender of patient | - | |
| **4** | Type | I | Spinner | - | User choose type of patient | - | |
| **5** | Reset | I/O | Button | - | Reset all filled fields to default | - | |
| **6** | OK | I/O | Button | - | Save patient information into database | - | |

#### 4.3.2.4 Health judgment

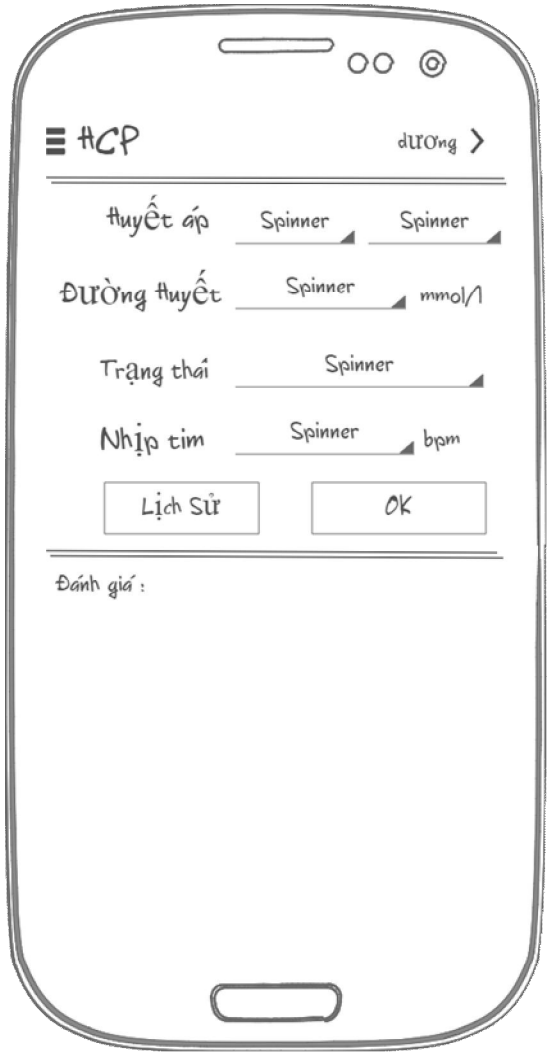



Figure 4-28 Health judgment screen prototype

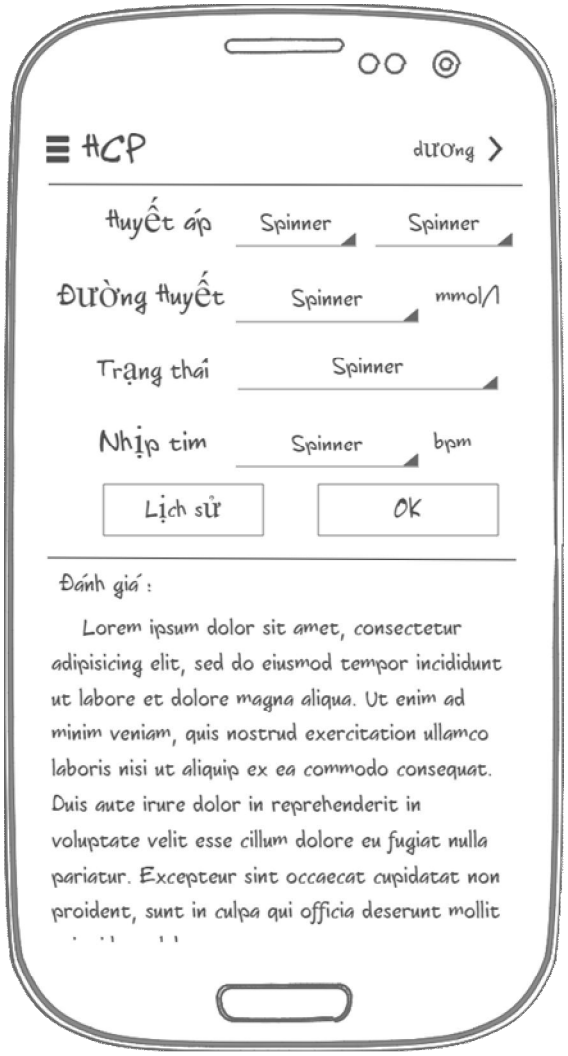

HCP
dương
Huyết áp
Spinner
Spinner
Đường huyết
Spinner
mmol/l
Trạng thái
Spinner
Nhịp tim
Spinner
bpm
Lịch sử
OK
Đánh giá :
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit

Figure 4-29 Health judgment screen after submit information prototype

TABLE 4-8 Health judgment screen description

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | | **Action bar** | | | |
| **1** | Refer main screen. | | | | | | |
| | | | | **Main content** | | | |
| **1** | Blood pressure | I | Spinner | - | User choose blood pressure range | - | |
| **2** | Glycemic | I | Spinner | - | User choose glycemic range | - | |
| **3** | Status | I | Spinner | - | User choose status when (Before/after have meal) | - | |
| **4** | Heart beat | I | Spinner | - | User choose heart beat range | - | |
| **5** | History | I/O | Button | - | List of all judgment user made | - | |
| **6** | OK | I/O | Button | - | Save patient information into database | - | |
| **7** | Judgment | O | Label | - | Application display judgment based on index entered by user | - | |

#### 4.3.2.5 Recently status

##### 4.3.2.5.1 Recently status list

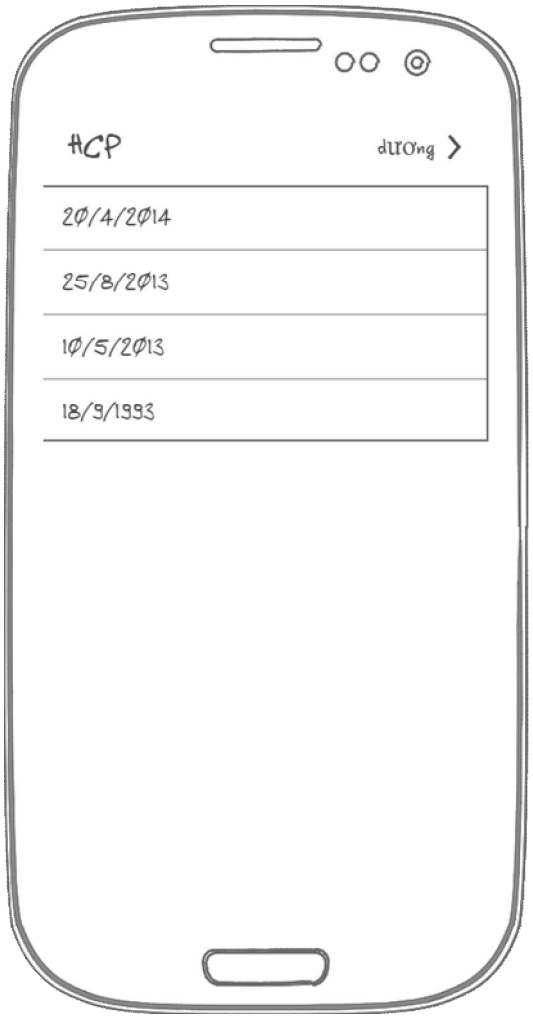



Figure 4-30 Recently status screen prototype

**TABLE 4-9 Recently status screen description**

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | | **Action bar** | | | |
| **1** | Refer Main screen | | | | | | |
| | | | | **Main content** | | | |
| **1** | Judgment date | O | List view | - | List of all date took judgment is displayed | - | |

#### 4.3.2.5.2 Recently status details

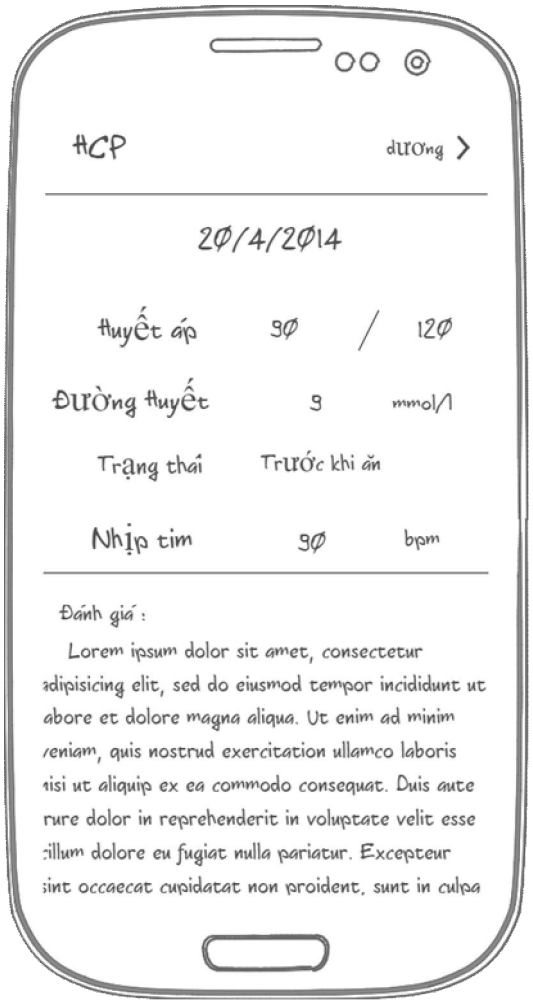

HCP
dương
20/4/2014
Huyết áp
90
120
Đường huyết
9
mmol/l
Trạng thái
Trước khi ăn
Nhịp tim
90
bpm
Đánh giá :
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut abore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute rure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa

Figure 4-31 Recently status detail screen prototype

**TABLE 4-10 Recently status detail screen description**

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | | | **Action bar** | | |
| **1** | Refer Main screen | | | | | | |
| | | | | | **Main content** | | |
| **1** | Blood pressure | O | Label | - | Blood pressure entered by user is displayed | - | |
| **2** | Glycemic | O | Label | - | Glycemic entered by user is displayed | - | |
| **3** | Status | O | Label | - | Status entered by user is displayed | - | |
| **4** | Heart beat | O | Label | - | Heart beat entered by user is displayed | - | |
| **5** | Judgment | O | Label | - | Content of judgment given by application | - | |

#### 4.3.2.6 Reminder

##### 4.3.2.6.1 Reminder list

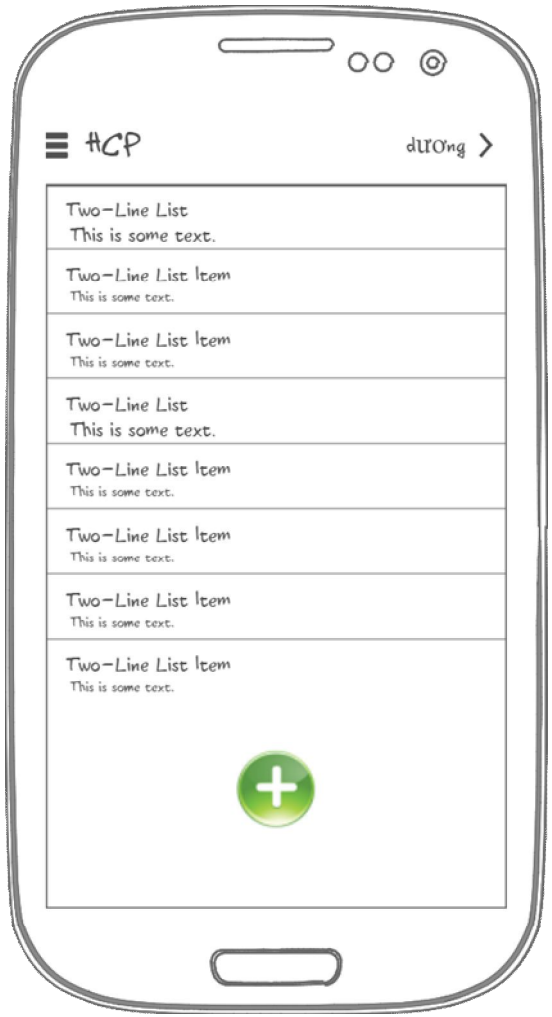



Figure 4-32 Reminder list screen prototype

TABLE 4-11 Reminder list screen description

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | | | **Action bar** | | |
| **1** | Refer Main screen | | | | | | |
| | | | | | **Main content** | | |
| 1 | Reminders | O | List view | - | List of all reminders related to current user on application is displayed | - | |
| 2 | | O | Button | | Add new reminder screen is displayed | - | |
| 3 | Today or All Reminder | O | Spinner | | List Today reminder or all reminder displayed | - | |

#### 4.3.2.6.2 Add new reminder

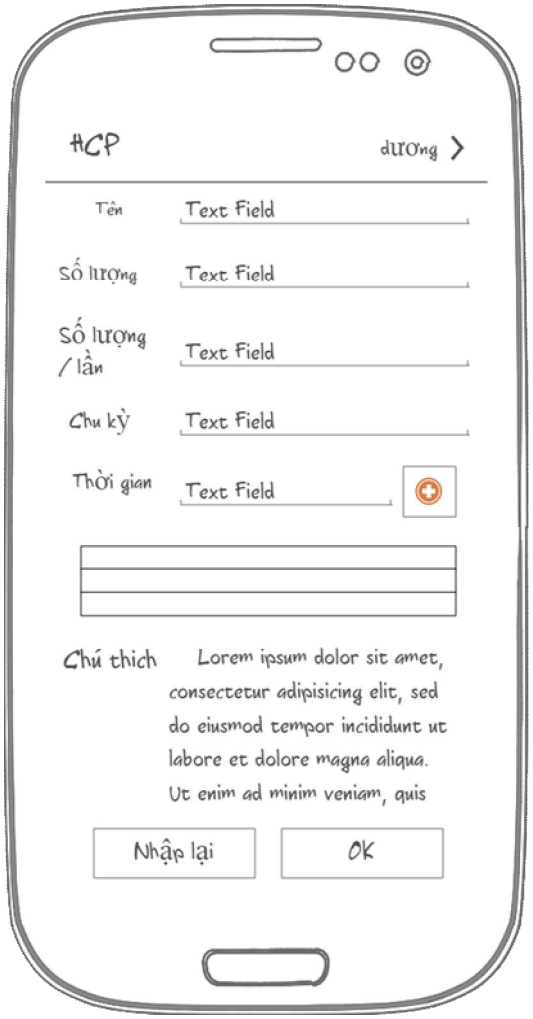



Figure 4-33 Add new reminder screen prototype

**TABLE 4-12 Add new reminder screen description**

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | **Action bar** | | | | |
| **1** | Refer Main screen | | | | | | |
| | | | **Main content** | | | | |
| **1** | Medicine name | I | Text field | 30 | Name of medicine | Must be filled and not have special character (@!~#$%^&*?<>:;) | |
| **2** | Total tablets | I | Text field | 3 | Total tablets of medicine | Must be filled and positive | Input number only |
| **3** | Number tablets per time | I | Text field | 2 | Number tablets patient have to take each time | Must be filled and positive | Input number only |
| **4** | Daily interval | I | Number picker | - | The daily interval of the reminder | Must be filled | Input number only |
| **5** | Time | I | Time picker | - | Time user have to take medicine | Must be filled | Can't be input by type |
| **6** | Time table | O | Table layout | - | All the times user inserted | - | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **7** | Note | I | Text field | - | Some more information that user might want to add about the reminder | - |
| **8** | Reset | I/O | Button | - | Reset all fields to default | - |
| **9** | OK | I/O | Button | - | Complete a reminder & save into database | - |

#### 4.3.2.6.3 Reminder details

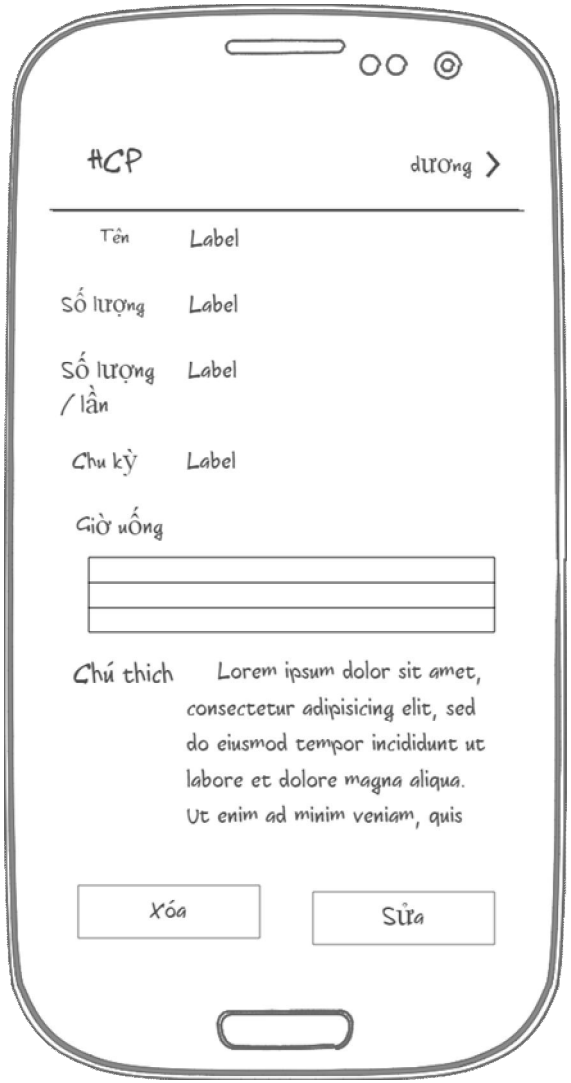



Figure 4-34 Reminder details screen prototype

TABLE 4-13 Reminder details screen description

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | **Action bar** | | | | |
| **1** | Refer Main screen | | | | | | |
| | | | **Main content** | | | | |
| **1** | Medicine name | O | Label | - | Name of medicine | - | |
| **2** | Total tablets | O | Label | - | Total tablets of medicine | - | |
| **3** | Number tablets per time | O | Label | - | Number tablets patient have to take each time | - | |
| **4** | Daily interval | O | Label | - | The daily interval of the reminder | - | |
| **5** | Time | O | Label | - | Time user have to take medicine | - | |
| **6** | Time table | O | Table layout | - | All the times user inserted | - | |
| **7** | Note | O | Label | - | Some more information that user might want to add | - | |
| **8** | Delete | I/O | Button | - | Delete this reminder | - | |
| **9** | Edit | I/O | Button | - | Edit this reminder | - | |

#### 4.3.2.6.4 Edit reminder

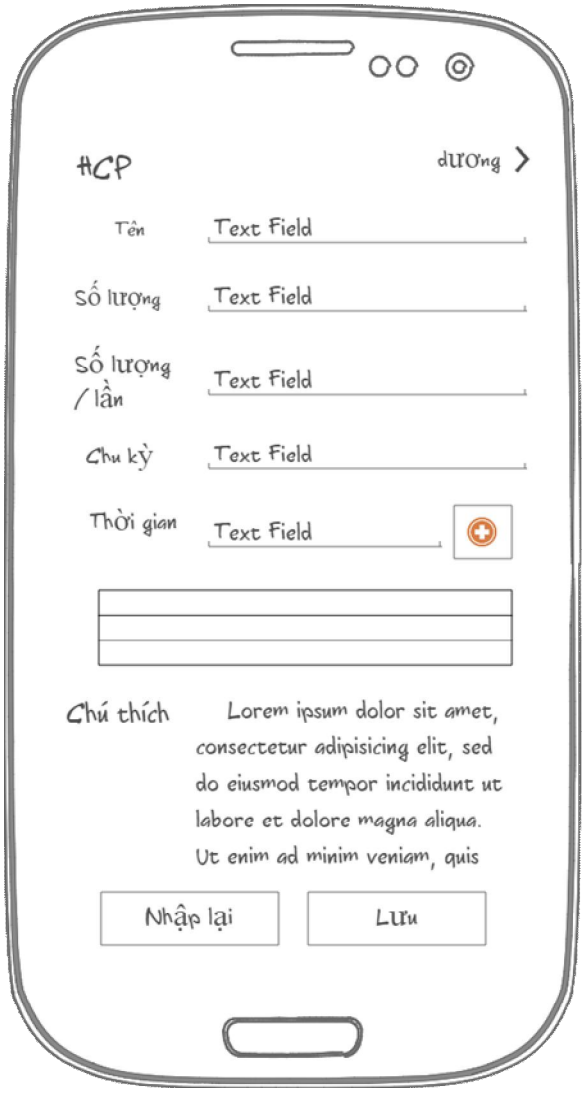



Figure 4-35 Edit reminder screen prototype

TABLE 4-14 Edit reminder screen description

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | **Action bar** | | | | |
| **1** | Refer Main screen | | | | | | |
| | | | **Main content** | | | | |
| **1** | Medicine name | I | Text field | 30 | Name of medicine | Must be filled and not have special character (@!~#$%^&*?<>:;) | |
| **2** | Total tablets | I | Text field | 3 | Total tablets of medicine | Must be filled and positive | Input number only |
| **3** | Number tablets per time | I | Text field | 2 | Number tablets patient have to take each time | Must be filled and positive | Input number only |
| **4** | Daily interval | I | Number picker | - | The daily interval of the reminder | Must be filled | Input number only |
| **5** | Time | I | Time picker | - | Time user have to take medicine | Must be filled | Can't be input by type |
| **6** | Time table | O | Table layout | - | All the times user inserted | - | |
| **7** | Note | I | Text field | - | Some more information that user might want to add about the reminder | - | |

| | | | | | | |
|---|---|---|---|---|---|---|
| **8** | Reset | I/O | Button | - | Reset all fields to default | - |
| **9** | OK | I/O | Button | - | Complete a reminder & save into database | - |

#### 4.3.2.6.5 Hospital

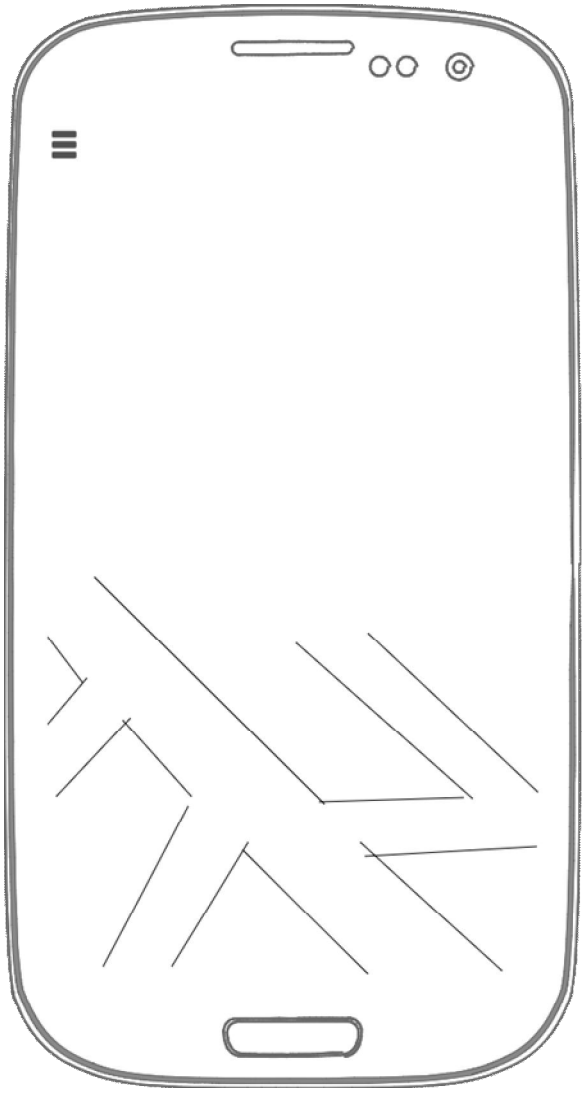


Figure 4-36 Hospital screen prototype

TABLE 4-15 Hospital screen description

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | | **Action bar** | | | |
| **1** | Refer Main screen | | | | | | |
| | | | | **Main content** | | | |
| **1** | Map | O | | | | | |

#### 4.3.2.6.6 Help

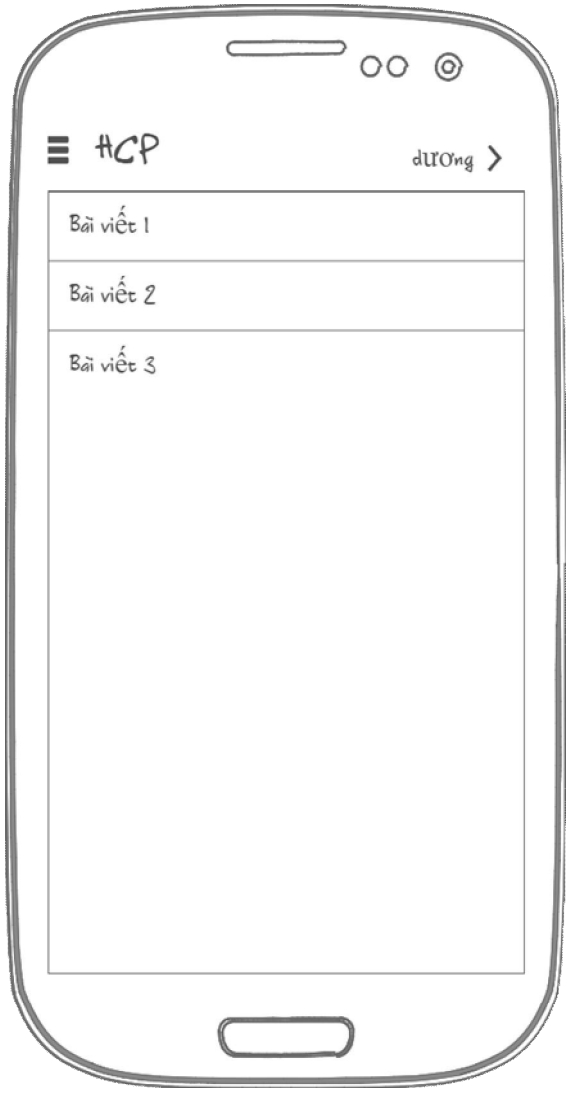



Figure 4-37 Help screen prototype

TABLE 4-16 Help screen description

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | | **Action bar** | | | |
| **1** | Refer Main screen | | | | | | |
| | | | | **Main content** | | | |
| 1 | Advisory list | O | List view | - | List of all advisory is displayed | - | |

#### 4.3.2.6.7 Detail Help

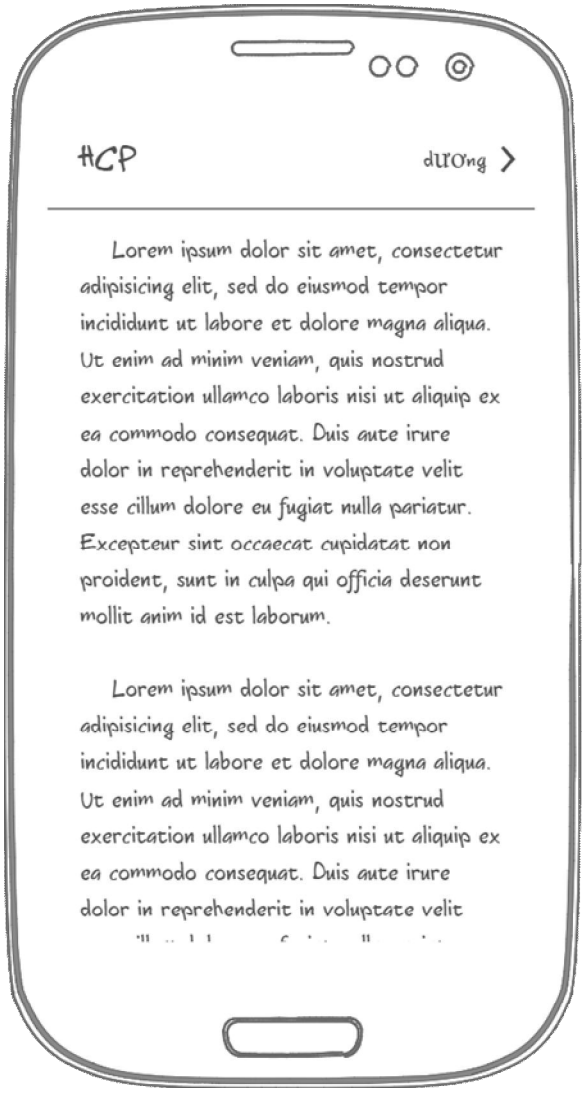

HCP
dUOng
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit

Figure 4-38 Help detail screen prototype

**TABLE 4-17 Help detail screen description**

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | **Action bar** | | | | |
| **1** | Refer Main screen | | | | | | |
| | | | **Main content** | | | | |
| 1 | Advisory content | O | Text View | - | Detail of advisory is displayed | - | |

#### 4.3.2.7 About us

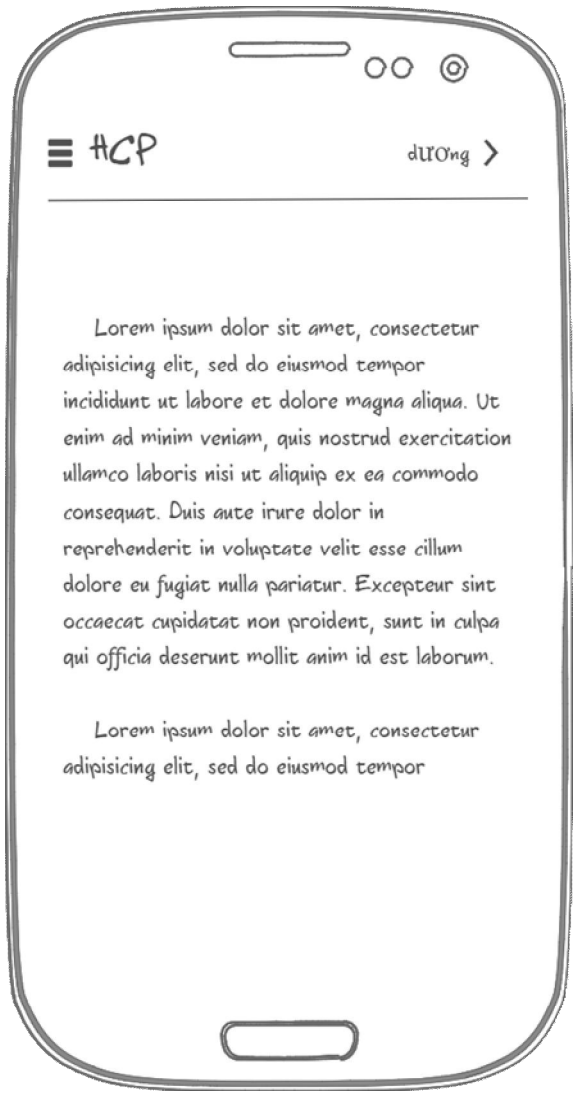

HCP
dương
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor

Figure 4-39 About us screen prototype

**TABLE 4-18 About us screen description**

| No | Element | I/O | Attribute | Digit count | Content of action | Input check | Note |
|---|---|---|---|---|---|---|---|
| | | | **Action bar** | | | | |
| **1** | Refer Main screen | | | | | | |
| | | | **Main content** | | | | |
| 1 | About | O | Text view | - | Show information about author | - | |

## 4.4 Data design

### 4.4.1 Entity Relationship Diagram

Figure 4-40 ERD diagram

TABLE 4-19 Database table list

| No | Table name | Description |
|---|---|---|
| **1** | Type | Store all information of standard person's type |
| **2** | Glycemic | Store all information of standard index of Glycemic and advices |
| **3** | HeartBeat | Store all information of standard index of Heart beat and advices |
| **4** | BloodPressure | Store all information of standard index of Blood pressure and advices |
| **5** | Patient | Store all information of patients |
| **6** | RecentlyStatus | Store all information about recently status inputted by user |
| **7** | Reminder | Store all information of reminder inputted by user |
| **8** | Medicine | Store name of all medicine inputted in reminder |
| **9** | Alarm | Store all information of the time for each reminder |
| **10** | Hospital | Store all information of the Hospital in Hanoi |
| **11** | Advisory | Store all information of advisory |

### 4.4.2 Table

Figure 4-41 ERD diagram

#### 4.4.2.1 Type table

| No | Field Name | Type | Null | Unique | P/F Key | Default | Description |
|---|---|---|---|---|---|---|---|
| **1** | TypeID | Integer | | x | P | | ID for each Type. TypeID is auto increment |
| **2** | LowerAge | Integer | | | | | The lower bound of the age for the Type |
| **3** | UpperAge | Integer | | | | | The upper bound of the age for the Type |
| **4** | Gender | Integer | | | | | The gender of the type |
| **5** | PersonType | Integer | | | | | The person type of the type |

#### 4.4.2.2 Glycemic table

| No | Field Name | Type | Null | Unique | P/F Key | Default | Description |
|---|---|---|---|---|---|---|---|
| **1** | GlycemicID | Integer | | x | P | | ID for each Glycemic record. GlycemicID is auto increment |
| **2** | TypeID | Integer | | | F | | The ID of the Type. TypeID is a foreign key from Type table |
| **3** | Glycemic | Text | | | | | The index of the Glycemic index for each record |
| **4** | Status | Integer | | | | | The status of patient (Before eating/ After eating) |
| **5** | Advice | Text | | | | | The advice base on the information of Type and Glycemic's index |

#### 4.4.2.3 HeartBeat table

| No | Field Name | Type | Null | Unique | P/F Key | Default | Description |
|---|---|---|---|---|---|---|---|
| **1** | HeartBeatID | Integer | | X | P | | ID for each Heart Beat record. HeartBeatID is auto increment |
| **2** | TypeID | Integer | | | F | | The ID of the Type. TypeID is a foreign key from Type table |
| **3** | HeartBeat | Text | | | | | The index of the Heart beat index for each record |
| **4** | Advice | Text | | | | | The advice base on the information of Type and Heart beat's index |

#### 4.4.2.4 BloodPressure table

| No | Field Name | Type | Null | Unique | P/F Key | Default | Description |
|---|---|---|---|---|---|---|---|
| **1** | BloodPressureID | Integer | | X | P | | ID for each Blood Pressure record. BloodPressureID is auto increment |
| **2** | TypeID | Integer | | | K | | The ID of the Type. TypeID is a foreign key from Type table |
| **3** | MinBloodPressure | Text | | | | | The index of the Min Blood Pressure index for each record |
| **4** | MaxBloodPressure | Text | | | | | The index of the Max Blood Pressure index for each record |
| **5** | Advice | Text | | | | | The advice base on the information of Type and Blood Pressure's index |

#### 4.4.2.5 Patient table

| No | Field Name | Type | Null | Unique | P/F Key | Default | Description |
|---|---|---|---|---|---|---|---|
| **1** | BloodPressureID | Integer | | X | P | | ID for each Blood Pressure record. BloodPressureID is auto increment |
| **2** | TypeID | Integer | | | K | | The ID of the Type. TypeID is a foreign key from Type table |
| **3** | MinBloodPressure | Text | | | | | The index of the Min Blood Pressure index for each record |
| **4** | MaxBloodPressure | Text | | | | | The index of the Max Blood Pressure index for each record |
| **5** | Advice | Text | | | | | The advice base on the information of Type and Blood Pressure's index |

#### 4.4.2.6 RecentlyStatus table

| No | Field Name | Type | Null | Unique | P/F Key | Default | Description |
|---|---|---|---|---|---|---|---|
| **1** | RecentlyStatusID | Integer | | x | P | | ID for each Recently Status. RecentlyStatusID is auto increment |
| **2** | PatientID | Integer | | | F | | The ID of patient whom this status belong to. PatientID is foreign key from Patient table |
| **3** | Date | Text | | | | | The date when the user input the status |
| **4** | Time | Text | | | | | The time when the user input the status |
| **5** | Glycemic | Text | | | | | The Glycemic's index of patient's status which user input on that time |
| **6** | Status | Integer | | | | | The status of Patient that time (Before eating / After eating) |
| **7** | HeartBeat | Text | | | | | The Heart beat's index of patient's status which user input on that time |
| **8** | MinBloodPressure | Text | | | | | The Min Blood Pressure's index of patient's status which user input on that time |
| **9** | MaxBloodPressure | Text | | | | | The Max Blood Pressure's index of patient's status which user input on that time |
| **10** | Advice | Text | | | | | The advice was given based on the information user inputted |

#### 4.4.2.7 Reminder table

| No | Field Name | Type | Null | Unique | P/F Key | Default | Description |
|---|---|---|---|---|---|---|---|
| **1** | ReminderID | Integer | | x | P | | ID for each Reminder. ReminderID is auto increment |
| **2** | Name | Text | | | | | The name of the reminder |
| **3** | Total | Integer | | | | | The total tablet of the reminder |
| **4** | NumPerTime | Integer | | | | | The number of tablet the patient will take each time |
| **5** | DailyInterval | Integer | | | | | The daily interval of the reminder |
| **6** | NextDay | Date | | | | | The next day that the patient will take the medicine |
| **7** | StartDay | Date | | | | | The day that the reminder will start being alarmed |
| **8** | Note | Text | x | | | | Some more information that user might want to add about the reminder |
| **9** | PatientID | Integer | | | F | | The ID of patient whom this reminder belong to. PatientID is foreign key from Patient table |

#### 4.4.2.8 Medicine table

| No | Field Name | Type | Null | Unique | P/F Key | Default | Description |
|---|---|---|---|---|---|---|---|
| **1** | MedicineID | Integer | | x | P | | ID for each medicine. MedicineID is auto increment |
| **2** | Name | Text | | | | | The name of the medicine |

#### 4.4.2.9 Alarm table

| No | Field Name | Type | Null | Unique | P/F Key | Default | Description |
|---|---|---|---|---|---|---|---|
| **1** | AlarmID | Integer | | X | P | | ID for each Alarm. AlarmID is auto increment |
| **2** | ReminderID | Integer | | | F | | The ID of the reminder which the time belong to. ReminderID is a foreign key from Reminder table |
| **3** | MedicineID | Integer | | | K | | The ID of the medicine which the time belong to. MedicineID is a foreign key from Reminder table |
| **4** | Time | Time | | | | | The time of the reminder |

#### 4.4.2.10 Hospital table

| No | Field Name | Type | Null | Unique | P/F Key | Default | Description |
|---|---|---|---|---|---|---|---|
| **1** | HospitalID | Integer | | x | P | | ID for each Hospital. HospitalID is auto increment |
| **2** | Name | Text | | | | | The name of the hospital |
| **3** | Information | Text | x | | | | Some basic information of the hospital |
| **4** | Latitude | Float | | | | | The latitude of the hospital's address |
| **5** | Longitude | Float | | | | | The longitude of the hospital's address |

#### 4.4.2.11 Advisory table

| No | Field Name | Type | Null | Unique | P/F Key | Default | Description |
|---|---|---|---|---|---|---|---|
| **1** | AdvisoryID | Integer | | x | P | | ID for each Advisory. AdvisoryID is auto increment |
| **2** | Title | Nvarchar | | | | | The title of the advisory |
| **3** | AdvisoryContent | Nvarchar | x | | | | The content of the advisory |

# 5 Software source code

## 5.1 Application

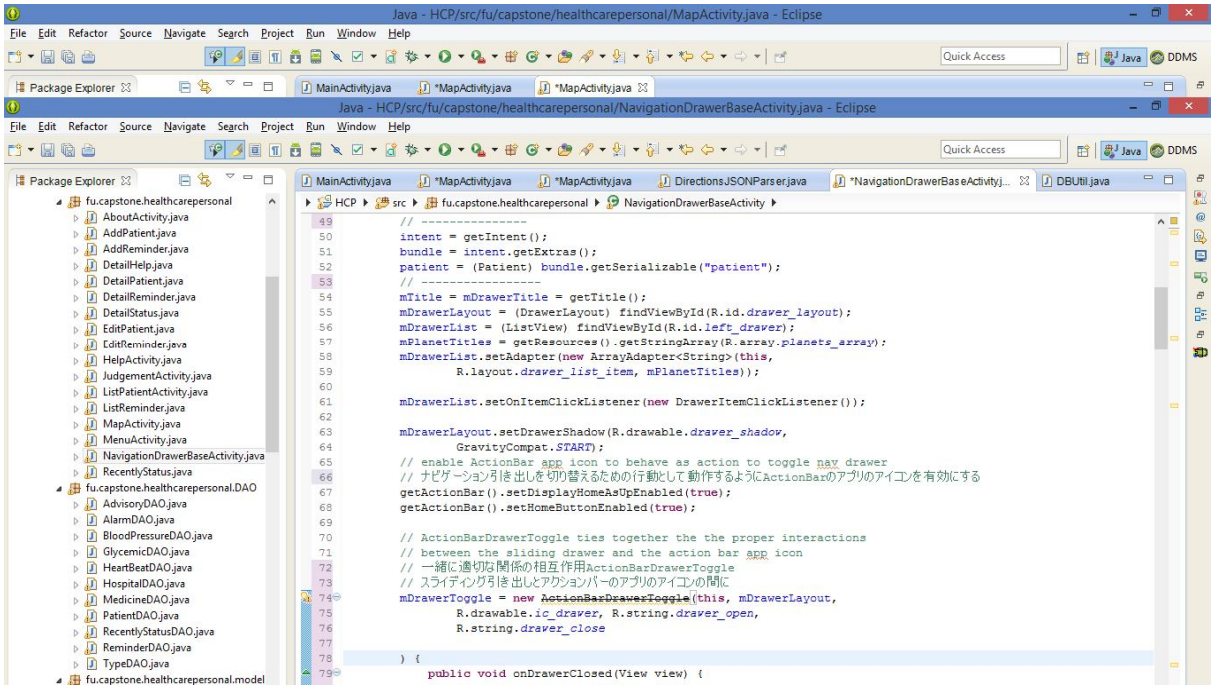



Figure 5-1 Source code 1

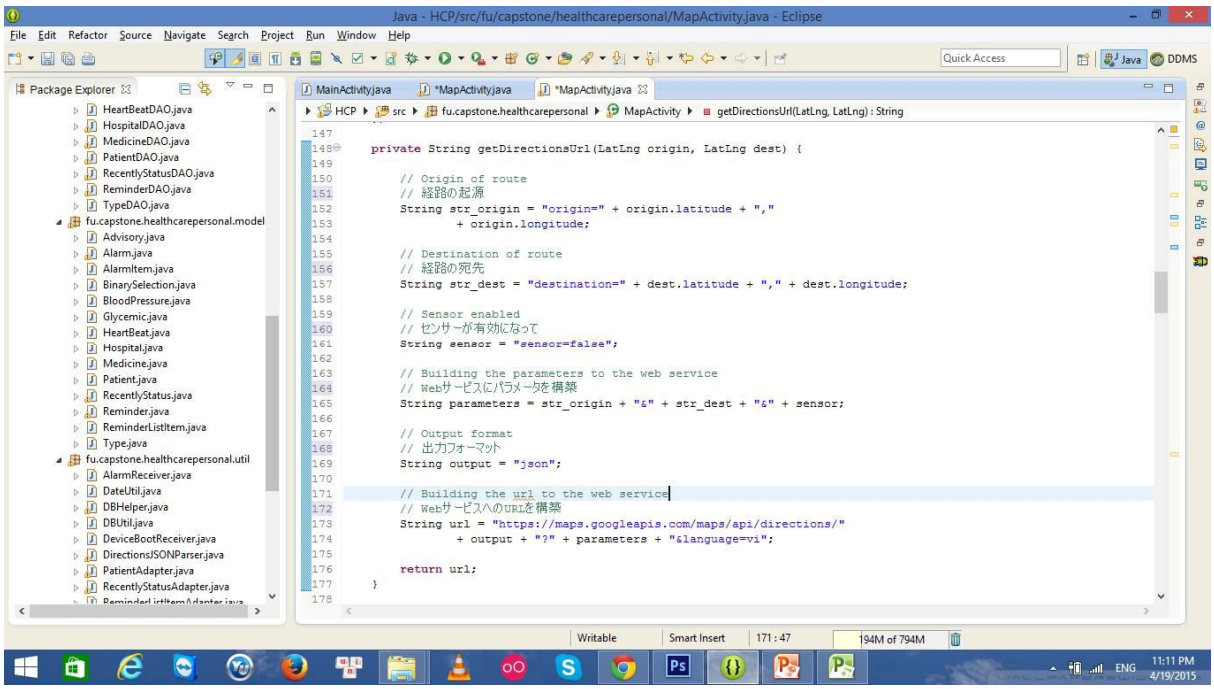



Figure 5-2 Source code 2

## 5.2 Database

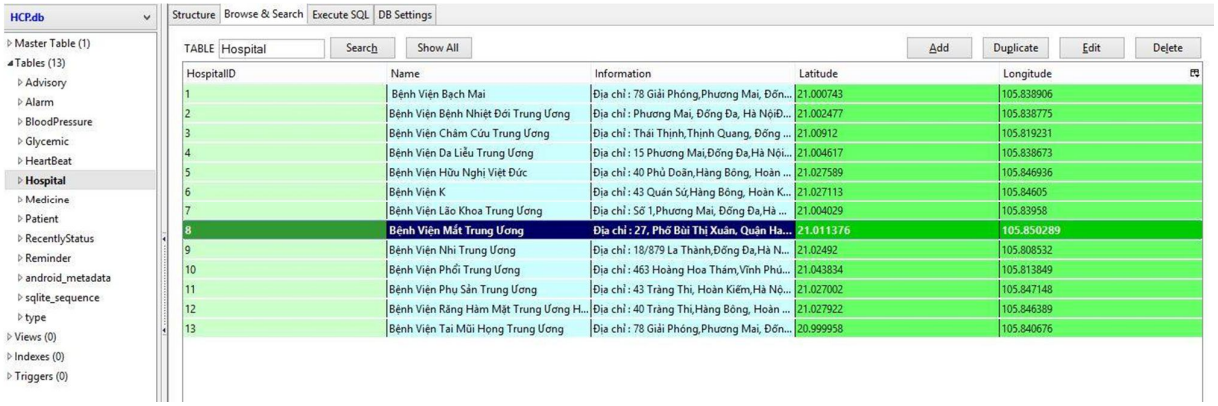

HCP.db
Structure
Browse & Search
Execute SQL
DB Settings
Master Table (1)
Tables (13)
Advisory
Alarm
BloodPressure
Glycemic
HeartBeat
Hospital
Medicine
Patient
RecentlyStatus
Reminder
android_metadata
sqlite_sequence
type
Views (0)
Indexes (0)
Triggers (0)
TABLE Hospital
Search
Show All
Add
Duplicate
Edit
Delete
HospitalID
Name
Information
Latitude
Longitude
1 Bệnh Viện Bạch Mai Địa chỉ : 78 Giải Phóng,Phương Mai, Đốn... 21.000743 105.838906
2 Bệnh Viện Bệnh Nhiệt Đới Trung Ương Địa chỉ : Phương Mai, Đống Đa, Hà NộiĐ... 21.002477 105.838775
3 Bệnh Viện Châm Cứu Trung Ương Địa chỉ : Thái Thịnh,Thịnh Quang, Đống ... 21.00912 105.819231
4 Bệnh Viện Da Liễu Trung Ương Địa chỉ : 15 Phương Mai,Đống Đa,Hà Nội... 21.004617 105.838673
5 Bệnh Viện Hữu Nghị Việt Đức Địa chỉ : 40 Phủ Doãn,Hàng Bông, Hoàn ... 21.027589 105.846936
6 Bệnh Viện K Địa chỉ : 43 Quán Sứ,Hàng Bông, Hoàn K... 21.027113 105.84605
7 Bệnh Viện Lão Khoa Trung Ương Địa chỉ : Số 1,Phương Mai, Đống Đa,Hà ... 21.004029 105.83958
8 Bệnh Viện Mắt Trung Ương Địa chỉ : 27, Phố Bùi Thị Xuân, Quận Ha... 21.011376 105.850289
9 Bệnh Viện Nhi Trung Ương Địa chỉ : 18/879 La Thành,Đống Đa,Hà N... 21.02492 105.808532
10 Bệnh Viện Phổi Trung Ương Địa chỉ : 463 Hoàng Hoa Thám,Vĩnh Phú... 21.043834 105.813849
11 Bệnh Viện Phụ Sản Trung Ương Địa chỉ : 43 Tràng Thi, Hoàn Kiếm,Hà Nộ... 21.027002 105.847148
12 Bệnh Viện Răng Hàm Mặt Trung Ương H... Địa chỉ : 40 Tràng Thi,Hàng Bông, Hoàn ... 21.027922 105.846389
13 Bệnh Viện Tai Mũi Họng Trung Ương Địa chỉ : 78 Giải Phóng,Phương Mai, Đốn... 20.999958 105.840676

Figure 5-3 Database Hospital table

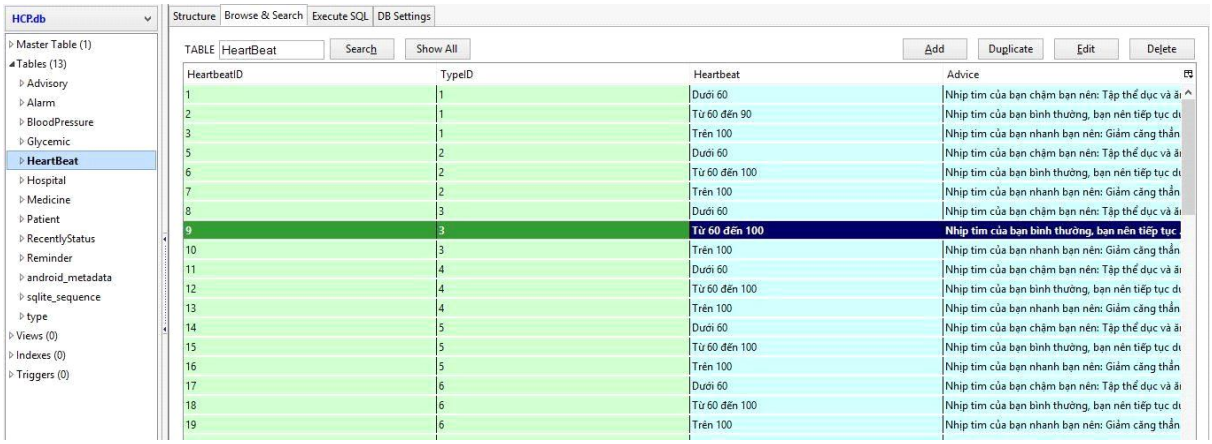

HCP.db
Structure
Browse & Search
Execute SQL
DB Settings
Master Table (1)
Tables (13)
Advisory
Alarm
BloodPressure
Glycemic
HeartBeat
Hospital
Medicine
Patient
RecentlyStatus
Reminder
android_metadata
sqlite_sequence
type
Views (0)
Indexes (0)
Triggers (0)
TABLE HeartBeat
Search
Show All
Add
Duplicate
Edit
Delete
HeartbeatID
TypeID
Heartbeat
Advice
1 1 Dưới 60 Nhịp tim của bạn chậm bạn nên: Tập thể dục và ă
2 1 Từ 60 đến 90 Nhịp tim của bạn bình thường, bạn nên tiếp tục d
3 1 Trên 100 Nhịp tim của bạn nhanh bạn nên: Giảm căng thẳn
5 2 Dưới 60 Nhịp tim của bạn chậm bạn nên: Tập thể dục và ă
6 2 Từ 60 đến 100 Nhịp tim của bạn bình thường, bạn nên tiếp tục d
7 2 Trên 100 Nhịp tim của bạn nhanh bạn nên: Giảm căng thẳn
8 3 Dưới 60 Nhịp tim của bạn chậm bạn nên: Tập thể dục và ă
9 3 Từ 60 đến 100 Nhịp tim của bạn bình thường, bạn nên tiếp tục
10 3 Trên 100 Nhịp tim của bạn nhanh bạn nên: Giảm căng thẳn
11 4 Dưới 60 Nhịp tim của bạn chậm bạn nên: Tập thể dục và ă
12 4 Từ 60 đến 100 Nhịp tim của bạn bình thường, bạn nên tiếp tục d
13 4 Trên 100 Nhịp tim của bạn nhanh bạn nên: Giảm căng thẳn
14 5 Dưới 60 Nhịp tim của bạn chậm bạn nên: Tập thể dục và ă
15 5 Từ 60 đến 100 Nhịp tim của bạn bình thường, bạn nên tiếp tục d
16 5 Trên 100 Nhịp tim của bạn nhanh bạn nên: Giảm căng thẳn
17 6 Dưới 60 Nhịp tim của bạn chậm bạn nên: Tập thể dục và ă
18 6 Từ 60 đến 100 Nhịp tim của bạn bình thường, bạn nên tiếp tục d
19 6 Trên 100 Nhịp tim của bạn nhanh bạn nên: Giảm căng thẳn

Figure 5-4 Database Heart beat table

# 6 Software Testing

## 6.1 Test Strategy

We use V-model to implement testing process.

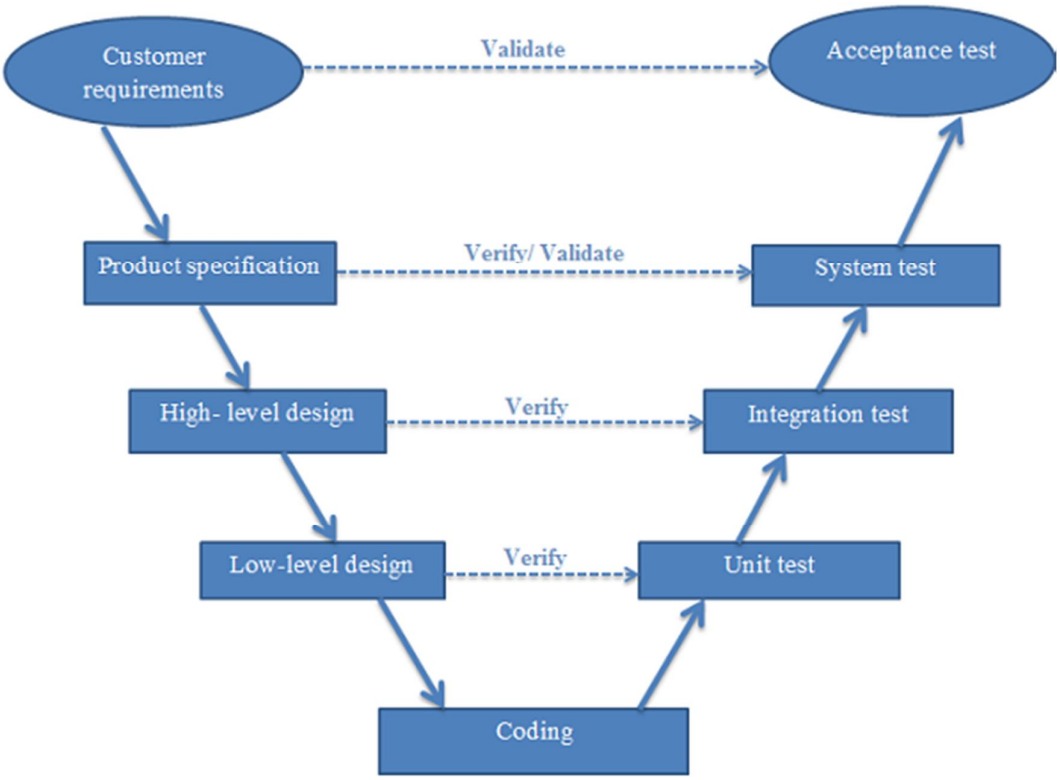



Figure 6-1 V-model

Due to requirement of project must be always suitable with user, we choose V-model to implement testing process. With V-Model, software development is separated into two appropriate phase's groups: development and testing. In this model, the verification and validation will be done side by side. It emphasizes the strict process flow to develop a quality product. The errors occurred in any phase will be corrected in that phase. Proactive defect tracking defects which are found at early stages even may be in the development phase before application being tested (Unit test).

Testing progress is divided to 4 phases include: Unit test, Integration test, System test and Acceptance test.

Because the project size is small, and not have customer, we decided not to execute acceptance test.

## 6.2 Test plan

### 6.2.1 Test type

#### 6.2.1.1 Function Testing

| | |
|---|---|
| **Test Objective** | Ensure proper target-of-test functionality, including navigation, data entry, processing, and retrieval. |
| **Technique** | Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following:<br>- The expected results occur when valid data is used.<br>- The appropriate error or warning messages are displayed when invalid data is used.<br>- Each business rule is properly applied.<br>- Use Test tool. |
| **Completion Criteria** | - All planned tests have been executed.<br>- All identified defects have been addressed and closed. |
| **Special Considerations** | Identify or describe those items or issues (internal or external) that impact the implementation and execution of function test. |

#### 6.2.1.2 User Interface Testing

| | |
|---|---|
| **Test Objective** | Verify the following:<br>Navigation through the target-of-test properly reflects business functions and requirements, including navigation bar, all of pages.<br>Objects and characteristics, such as menus, size, position, state, and focus conform to standards. |
| **Technique** | Create or modify tests for each page to verify proper navigation and object states for each application and objects. |
| **Completion Criteria** | Each page successfully verified to remain consistent with benchmark version or within acceptable standard. |
| **Special Considerations** | Not all properties for custom and third party objects can be accessed. |

#### 6.2.1.3 Performance Testing

| | |
|---|---|
| **Test Objective** | Verify performance behaviors for designated transactions or business functions under the following conditions:<br>- Normal anticipated workload( Normal time to load a page)<br>- Anticipated worst case workload (Longest time to load a page) |

| | |
|---|---|
| **Technique** | Use Test Procedures developed for Function or Business Cycle Testing.<br>Modify data files to increase the number of transactions or the scripts to increase the number of iterations each transaction occurs.<br>Scripts should be run on one machine (best case to benchmark single user, single transaction) and be repeated with multiple clients (virtual or actual). |
| **Completion Criteria** | Single Transaction or single user: Successful completion of the test scripts without any failures and within the expected or required time allocation per transaction.<br>Multiple transactions or multiple users: Successful completion of the test scripts without any failures and within acceptable time allocation. |
| **Special Considerations** | Comprehensive performance testing includes having a background workload on the server.<br>There are several methods that can be used to perform this, including:<br>"Drive transactions" directly to the server, usually in the form of Structured Query Language (SQL) calls.<br>Create "virtual" user load to simulate many clients, usually several hundred. Remote Terminal Emulation tools are used to accomplish this load. This technique can also be used to load the network with "traffic".<br>Use multiple physical clients, each running test scripts to place a load on the system.<br>Performance testing should be performed on a dedicated machine or at a dedicated time. This permits full control and accurate measurement.<br>The databases used for Performance Testing should be either actual size or scaled equally. |

### 6.2.2 Requirement for test

#### *6.2.2.1 Feature to be tested*

Developers will be performed unit test to ensure that all functions are working as expected.

The test team will be undertaking the responsibility of testing and validating the operation of the part that developed. The functions developed require completing design and execution of function tests to ensure proper coverage.

Following are items that will need to be tested by testers:

**TABLE 6-1 Feature to be tested**

| No. | Component Name | Feature Name | | Feature to be tested |
|---|---|---|---|---|
| **1** | Interface | Patient | "Thêm người dùng" screen | Check Display Element |
| **2** | | | "Danh sách người dùng" screen | Check Display Element |
| **3** | | | "Chi tiết người dùng" screen | Check Display Element |
| **4** | | | "Chỉnh sửa thông tin" screen | Check Display Element |
| **5** | | Main | Main screen | Check Display Element |
| **6** | | Navigator | Navigator screen | Check Display Element |
| **7** | | Judgment | "Đánh giá" screen | Check Display Element |
| **8** | | | "Tình trạng gần đây" screen | Check Display Element |
| **9** | | | "Chi tiết tình trạng" screen | Check Display Element |
| **10** | | Reminder | "Danh sách nhắc nhở" screen | Check Display Element |
| **11** | | | "Chi tiết lời nhắc nhở" screen | Check Display Element |
| **12** | | | "Thêm mới lời nhắc nhở" screen | Check Display Element |
| **13** | | | "Thay đổi lời nhắc nhở" screen | Check Display Element |
| **14** | | Hospital | "Hiển thị bệnh viện" screen | Check Display Element |
| **15** | | | Open Map screen | Check Display Element |
| **16** | | Setting | "Cài đặt" screen | Check Display Element |
| **17** | | Help | "Trợ giúp" screen | Check Display Element |
| **18** | | About us | "Về chúng tôi" screen | Check Display Element |
| **19** | Function | First time login | | Function work. |
| **20** | | Patient Management | View Patient List | Function work. |
| **21** | | | Add New Patient | Function work. |
| **22** | | | Switch Patient | Function work. |
| **23** | | | Patient Detail Information | Function work. |
| **24** | | | Edit Patient Information | Function work. |
| **25** | | | Delete Patient | Function work. |

| | | | | |
|---|---|---|---|---|
| **26** | | Health judgment | Health Judgment | Function work. |
| **27** | | | Recently Status | Function work. |
| **28** | | Reminder | View Reminders List | Function work. |
| **29** | | | Add New Reminder | Function work. |
| **30** | | | Reminder Detail Information | Function work. |
| **31** | | | Edit Reminder Detail | Function work. |
| **32** | | | Delete Reminder | Function work. |
| **33** | | Hospital | Hospital | Function work. |
| **34** | | Advisory | Advisory | Function work. |
| **35** | Performance | | | Response Time |

#### *6.2.2.2 Acceptance Test Criteria*

List of criteria to define what levels of test quality are sufficient to move to the next testing phase:

- Test coverage: 100%
- Successful Test coverage: 90%
- Number of Test cases (Unit/Integration/System Test cases): 568 TC
- Number of defects/Weighted defects: 3~4 defects/KLOC

List of criteria which are based on to accept the products:

- Number of UTC/KLOC: 30~50 UTC/KLOC
- Number or Weighted: 6~8 defects/KLOC
- Statement coverage: 90%
- Branch coverage: 100%
- Path coverage: 100%

### 6.2.3 Test stage

| Type of Tests | Stage of Test | | | |
|---|---|---|---|---|
| | Unit | Integration | System | Acceptance |
| **Function Test** | X | X | X | X |
| **User Interface test** | X | | X | |
| **Performance Tests** | X | | X | |

### 6.2.4 Create test cases plan

- **Unit Test**

| Date | Feb -15 | | | | | Total |
|---|---|---|---|---|---|---|
| | 14/2 | 23/2 | 24/2 | 25/2 | 26/2 | |
| **AnhDT** | 14 | 14 | 14 | 14 | 14 | 80 |
| **HuongPT** | 16 | 16 | 16 | 16 | 16 | 90 |
| **DuongDT** | 14 | 14 | 14 | 14 | 14 | 80 |
| **Sum** | 44 | 44 | 44 | 44 | 44 | 250 |

- **Function**

| Date | Feb -15 | | | | | | | Mar - 15 | | | | | | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 14/2 | 23/2 | 24/2 | 25/2 | 26/2 | 27/2 | 28/2 | 2/3 | 3/3 | 4/3 | 5/3 | 6/3 | 7/3 | |
| **ThienLK** | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 130 |
| **LinhNC** | 11 | 11 | 11 | 11 | 11 | 11 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 150 |
| **Sum** | 21 | 21 | 21 | 21 | 21 | 21 | 22 | 22 | 22 | 22 | 22 | 22 | 22 | 280 |

### 6.2.5 Execute test plan

- **Test UT**

| Date | Mar -15 | | | | | | | | | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| | 10/3 | 11/3 | 12/3 | 13/3 | 14/3 | 16/3 | 17/3 | 18/3 | 19/3 | |
| **Execute Test (TC)** | 30 | 30 | 30 | 30 | 30 | 25 | 25 | 25 | 25 | 250 |
| **AnhDT** | X | X | X | X | X | X | X | X | X | |
| **DuongDT** | x | X | X | X | X | X | X | X | x | |
| **HuongPT** | X | X | X | X | X | X | X | X | x | |
| **Bug** | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 |

- **Test Function**

| Date | Round 1 | | | Round 2 | Total |
|---|---|---|---|---|---|
| | Mar-15 | | | Apr-15 | |
| | 21/3 | 23/3 | 24/3 | 3/4 | |
| **Execute Test (TC)** | 85 | 90 | 105 | 280 | 360 |
| **ThienLK** | x | x | x | x | |
| **LinhNC** | x | x | x | x | |
| **Bug** | 8 | 8 | 8 | 1 | 25 |
| **Method** | Patient, Reminder | Use case, GUI | Common, Performance. | All | |

## 6.3 Testing Environments

### 6.3.1 Software

| Category | Software name | Version |
|---|---|---|
| **Operating System** | Android 4.3 | Jelly Bean |
| | Android 4.4 | KitKat |
| **Development framework** | Microsoft .NET | 4.0 |
| **Development tools** | Eclipse | 2010 |
| | Notepad++ | Newest |
| **Database tools** | SQLite manager for Firefox | Newest |
| **Support tools** | Firefox | Newest |
| **Testing tools** | JUnit | Newest |

### 6.3.2 Hardware

Android smart phone 1:

- Android 4.4 KitKat
- Intel Atom Z2520, Core Duo, 1.2GHz
- Install memory (RAM) : 1.00GB

Android smart phone 2:

- Android 4.3 Jelly Bean

- Intel Atom Z2520, Core Duo, 1.2GHz
- Install memory (RAM) : 1.00GB

## 6.4 Test Milestones

| Milestone Task | Resource | Start Date | Finish date |
|---|---|---|---|
| **Test plan** | Test leader | 2015/02/11 | 2015/02/12 |
| **Review Test plan** | Project Manager | 2015/02/13 | 2015/02/13 |
| **Create Unit Test case** | Developer | 2015/02/14 | 2015/02/18 |
| **Create test case for patient function** | Tester | 2015/02/14 | 2015/02/26 |
| **Create test case for reminder function** | Tester | 2015/02/14 | 2015/02/24 |
| **Create test case for common function** | Tester | 2015/02/27 | 2015/03/02 |
| **Create GUI TC** | Tester | 2015/03/03 | 2015/03/05 |
| **Create performance TC** | Tester | 2015/03/06 | 2015/03/07 |
| **Review test case** | Project Manager | 2015/03/09 | 2015/03/09 |
| **Test case approved** | Project Manager | 2015/03/10 | 2015/03/10 |
| **Test UT** | Developer | 2015/03/10 | 2015/03/19 |
| **Test round 1** | Tester | 2015/03/21 | 2015/03/24 |
| **Test round 2** | Tester | 2015/04/03 | 2015/04/03 |
| **Create test report** | Tester | 2015/04/07 | 2015/04/07 |

## 6.5 Test case

### 6.5.1 Unit test case

| Function Code | Add Patient | Function Name | Add new patient |
|---|---|---|---|
| Created By | AnhDT | Executed By | AnhDT |
| Lines of code | 60 | Lack of test cases | |
| Test requirement | Test insert new information of patient | | |

| Passed | Failed | Untested | N/A | B | Total Test Cases |
|---|---|---|---|---|---|
| 12 | 0 | 0 | 3 | 8 | 1 | 12 |

| | | UTCID01 | UTCID02 | UTCID03 | UTCID04 | UTCID05 | UTCID06 | UTCID07 | UTCID08 | UTCID09 | UTCID10 | UTCID11 | UTCID12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Condition | Precondition | | | | | | | | | | | | |
| | HCP app is opened | | | | | | | | | | | | |
| | Input name of patient | | | | | | | | | | | | |
| | null | O | | | | | | | | | | | |
| | abc! | | O | | | | | | | | | | |
| | abc 123 | | | O | | | | | | | | | |
| | abcabasbasbdasdbasdabdasdabdasdb | | | | O | | | | | | | | |
| | 1232435 | | | | | O | | | | | | | |
| | anh | | | | | | O | | | | | | |
| | Input year of birth | | | | | | | | | | | | |
| | null | | | | | | | O | | | | | |
| | -1993 | | | | | | | | O | | | | |
| | 1993 | | | | | | | | | O | | | |
| | 100 | | | | | | | | | | O | | |
| | abc | | | | | | | | | | | O | |
| | 1993# | | | | | | | | | | | | O |
| Confirm | Return | | | | | | | | | | | | |
| | Add successfully | | | O | | | O | | | O | | | |
| | Exception | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | Log message | | | | | | | | | | | | |
| | "Hãy hập tên bệnh nhân" | O | | | | | | | | | | | |
| | "Sai tên bệnh nhân" | | O | | O | O | | | | | | | |
| | "Sai năm sinh bệnh nhân"' | | | | | | | O | O | | O | O | O |
| Result | Type(N : Normal, A : Abnormal, B : Boundary) | A | A | N | B | A | N | A | A | N | A | A | A |
| | Passed/Failed | P | P | P | P | P | P | P | P | P | P | P | P |
| | Executed Date | 0 | 0 | 0 | 0 | 0 | 0 | | | | | | |
| | Defect ID | | | | | | | | | | | | |

Figure 6-2 Unit test case

### 6.5.2 Test use case

| Module Code | Use Case - Patient |
|---|---|
| Test requirement | Test use case of patient |
| Tester | |

| Pass | Fail | Untested | N/A | Number of Test cases |
|---|---|---|---|---|
| 0 | 0 | 21 | 0 | 21 |

| ID | Test Case Description | Test Case Procedure | Expected Output | Inter-test case Dependence | Result | Test date | Creator | Tester | Note |
|---|---|---|---|---|---|---|---|---|---|
| [Use Case - Patient-3] | Test "Add new patient" use case - "Tên" is blank. | 1. Tap on add button on "Danh sách bệnh nhân" screen.<br>2. Input information into fields at the screen with the following information:<br>- "Tên" field is blank.<br>- "Năm sinh": 1900.<br>- "Giới tính": "Nam"<br>- "Loại": "Người ít vận động"<br>3. Tap on "OK" button. | 1. "Thêm bệnh nhân" screen is displayed.<br>3. "Hãy nhập tên bệnh nhân" message is showed. | | | | ThienLK | | |
| [Use Case - Patient-4] | Test "Add new patient" use case - "Tên" exist special character. | 1. Tap on add button on "Danh sách bệnh nhân" screen.<br>2. Input information into fields at the screen with the following information:<br>- "Tên": "ABC@".<br>- "Năm sinh": 1999.<br>- "Giới tính": "Nam"<br>- "Loại": "Người ít vận động"<br>3. Tap on "OK" button. | 1. "Thêm bệnh nhân" screen is displayed.<br>3. "Tên không bao gồm kí tự đặc biệt như: @#$%^&*()!"':;{}\|\-_+=<>.,?/..." message is showed. | | | | ThienLK | | |
| [Use Case - Patient-5] | Test "Add new patient" use case - "Tên" is existed. | 1. Tap on add button on "Danh sách bệnh nhân" screen.<br>2. Input information into fields at the screen with the following information:<br>- "Tên": "ABC".<br>- "Năm sinh": 1993.<br>- "Giới tính": "Nam"<br>- "Loại": "Người ít vận động"<br>3. Tap on "OK" button. | 1. "Thêm bệnh nhân" screen is displayed.<br>3. "Tên bệnh nhân đã tồn tại." message is showed. | Application already have a patient name "ABC" | | | ThienLK | | |
| [Use Case - Patient-6] | Test "Add new patient" use case - Normal case. | 1. Tap on add button on "Danh sách bệnh nhân" screen.<br>2. Input information into fields at the screen with the following information: | 1. "Thêm bệnh nhân" screen is displayed.<br>3. Main screen is display with name of patient on Action bar. | | | | ThienLK | | |

Figure 6-3 Patient use case test case

| Module Code | *Use Case - Reminder* | | | |
|---|---|---|---|---|
| Test requirement | *Test use case of user* | | | |
| Tester | | | | |
| **Pass** | **Fail** | **Untested** | **N/A** | **Number of Test cases** |
| 48 | 22 | 0 | 0 | 70 |

| ID | Test Case Description | Test Case Procedure | Expected Output | Inter-test case Dependency | Result | Test date | Creator | Tester | Note |
|---|---|---|---|---|---|---|---|---|---|
| | **Add new reminder** | | | | | | | | |
| [Use Case - Reminder-1] | Test "Add new reminder" use case<br>- All fields are blank | 1. Tap on button Add in "Danh sách nhắc nhở" screen | 1. "Thêm lời nhắc nhở" screen is displayed | | Pass | | LinhNC | | |
| [Use Case - Reminder-2] | Test "Add new reminder" use case<br>- All fields are blank | 1. Tap on button Add in "Danh sách nhắc nhở" screen<br>2. Input information as follow:<br>- All fileds are blank<br>3. Tap on button OK | 1. "Thêm lời nhắc nhở" screen is displayed<br>3. Error message "Hãy nhập tên thuốc" is displayed | | Pass | | LinhNC | | |
| [Use Case - Reminder-3] | Test "Add new reminder" use case<br>- Test "Tên thuốc" field | 1. Tap on button Add in "Danh sách nhắc nhở" screen<br>2. Input information as follow:<br>- "Tên thuốc" is blank<br>- "Số lượng" is 1<br>- "Số lượng/lần" is 1<br>- "Chu kỳ" is 1<br>- "Thời gian" is 07:00<br>- "Chú thích" is blank<br>3. Tap on button OK | 1. "Thêm lời nhắc nhở" screen is displayed<br>3. Error message "Hãy nhập tên thuốc" is displayed | | Pass | | LinhNC | | |
| [Use Case - Reminder-4] | Test "Add new reminder" use case<br>- Test "Tên thuốc" field | 1. Tap on button Add in "Danh sách nhắc nhở" screen<br>2. Input information as follow:<br>- "Tên thuốc" have special character: "A@"<br>- "Số lượng" is 1<br>- "Số lượng/lần" is 1<br>- "Chu kỳ" is 1<br>- "Thời gian" is 07:00<br>- "Chú thích" is blank<br>3. Tap on button OK | 1. "Thêm lời nhắc nhở" screen is displayed<br>3. Error message "Hãy nhập tên thuốc" is displayed | | Fail | | LinhNC | | |
| [Use Case - Reminder-5] | Test "Add new reminder" use case<br>- Test "Tên thuốc" field | 1. Tap on button Add in "Danh sách nhắc nhở" screen<br>2. Input information as follow:<br>- "Tên thuốc" have number: "A1"<br>- "Số lượng" is 2 | 1. "Thêm lời nhắc nhở" screen is displayed<br>3. "Danh sách nhắc nhở" screen is displayed with new reminder "A1"<br>4. Details of reminder "A1" is displayed in "Chi tiết nhắc | | Pass | | LinhNC | | |

Figure 6-4 Reminder use case test case

### 6.5.3 GUI test case

| Module Code | *GUI* | | | |
|---|---|---|---|---|
| Test requirement | | | | |
| Tester | | | | |
| **Pass** | **Fail** | **Untested** | **N/A** | **Number of Test cases** |
| 86 | 18 | 0 | 0 | 104 |

| ID | Test Case Description | Test Case Procedure | Expected Output | Inter-test case Dependency | Result | Test date | Creator | Tester | Note |
|---|---|---|---|---|---|---|---|---|---|
| [GUI-15] | Test Navigation drawer Form - From main screen. | 1. Tap on navigation drawer symbol at main screen. | 1.Navigation drawer is displayed with the following informations:<br>- Name of application: "HCP".<br>- "Quản lý" item.<br>- "Đánh giá" item.<br>- "Tình trạng gần đây" item.<br>- "Nhắc nhở uống thuốc" item.<br>- "Hiển thị bệnh viện" item.<br>- "Cài đặt" item.<br>- "Trợ giúp" item.<br>- "Về chúng tôi" item. | | Pass | | ThienLK | | |
| [GUI-16] | Test Navigation drawer Form - From "Đánh giá" screen. | 1. Tap on navigation drawer symbol at "Đánh giá" screen. | 1.Navigation drawer is displayed with the following informations:<br>- Name of application: "HCP".<br>- "Quản lý" item.<br>- "Đánh giá" item.<br>- "Tình trạng gần đây" item.<br>- "Nhắc nhở uống thuốc" item.<br>- "Hiển thị bệnh viện" item.<br>- "Cài đặt" item.<br>- "Trợ giúp" item.<br>- "Về chúng tôi" item. | | Pass | | ThienLK | | |
| [GUI-17] | Test Navigation drawer Form - From "Tình trạng gần đây" screen. | 1. Tap on navigation drawer symbol at "Tình trạng gần đây" screen. | 1.Navigation drawer is displayed with the following informations:<br>- Name of application: "HCP".<br>- "Quản lý" item.<br>- "Đánh giá" item.<br>- "Tình trạng gần đây" item.<br>- "Nhắc nhở uống thuốc" item.<br>- "Hiển thị bệnh viện" item.<br>- "Cài đặt" item.<br>- "Trợ giúp" item.<br>- "Về chúng tôi" item. | | Pass | | ThienLK | | |
| [GUI-18] | Test Navigation drawer Form - From "Chi tiết tình trạng" | 1. Tap on navigation drawer symbol at "Chi tiết tình trạng" screen. | 1.Navigation drawer is displayed with the following informations: | | Pass | | ThienLK | | |

Figure 6-5 GUI test case

### 6.5.4 Performance test case

| Module Code | Performance | | | |
|---|---|---|---|---|
| Test requirement | Test perfomance of Android application | | | |
| Tester | | | | |
| Pass | Fail | Untested | N/A | Number of Test cases |
| 15 | 0 | 0 | 0 | 15 |

| ID | Test Case Description | Test Case Procedure | Expected Output | Inter-test case | Result | Test dat | Creator | Tester | Note |
|---|---|---|---|---|---|---|---|---|---|
| | Function D | | | | | | | | |
| [Performance-1] | Load Main screen | 1. Open Main screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-2] | Load Navigator Drawer screen | 1. Open Navigator Drawer | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-3] | Load "Add new patient" screen | 1. Open "Add new patient" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-4] | Load "Patient list" screen | 1. Open "Patient list" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-5] | Load "Patient detail" screen | 1. Open "Patient detail" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-6] | Load "Edit patient detail" screen | 1. Open "Edit patient detail" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-7] | Load "Health Judgment" screen | 1. Open "Health judgment" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-8] | Load "Recently status" screen | 1. Open "Recently status" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-9] | Load "Recently status detail" screen | 1.Open "Recently status detail" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-10] | Load "Reminder list" screen | 1.Open "Reminder list" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-11] | Load "Add new reminder" screen | 1.Open "Add new reminder" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-12] | Load "Reminder detail" screen | 1.Open "Reminder detail" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-13] | Load "Edit reminder detail" screen | 1.Open "Edit reminder detail" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-14] | Load "Hospital" screen | 1.Open "Hospital" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |
| [Performance-15] | Load "Advisory" screen | 1.Open "Advisory" screen | 1. <3 seconds | | Pass | | ThienLK | ThienLK | |

Figure 6-6 Performance test case

## 6.6 Test report

- **Unit Test**

| Date | Mar -15 | | | | | | | | | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| | 10/3 | 11/3 | 12/3 | 13/3 | 14/3 | 16/3 | 17/3 | 18/3 | 19/3 | |
| **Execute Test (TC)** | 25 | 24 | 21 | 19 | 38 | 32 | 40 | 33 | 38 | 270 |
| **AnhDT** | X | X | X | X | X | X | X | X | X | |
| **DuongDT** | X | X | X | X | X | X | X | X | x | |
| **HuongPT** | X | X | X | X | X | X | X | X | x | |
| **Bug** | 0 | 0 | 0 | 3 | 1 | 0 | 1 | 0 | 0 | 5 |

| No | Module code | Pass | Fail | Untested | N/A | Number of test cases |
|---|---|---|---|---|---|---|
| **1** | Unit Test | 258 | 12 | 0 | 0 | 270 |

- **Function Test**

| Date | Round 1 | | | Round 2 | Total |
|---|---|---|---|---|---|
| | Mar-15 | | | Apr-15 | |
| | 21/3 | 23/3 | 24/3 | 3/4 | |
| **Execute Test (TC)** | 93 | 114 | 77 | 284 | 568 |
| **ThienLK** | x | x | x | X | |
| **LinhNC** | x | x | x | X | |
| **Bug** | 6 | 3 | 0 | | 9 |
| **Method** | Patient, Reminder | Other, GUI | Common, Performance. | All | |

➢ **Round 1**

| No | Module code | Pass | Fail | Untested | N/A | Number of test cases |
|---|---|---|---|---|---|---|
| **1** | Use case | 5 | 3 | 0 | 0 | 8 |
| **2** | Use case – Patient | 15 | 5 | 3 | 0 | 23 |
| **3** | Use case – Reminder | 45 | 20 | 5 | 0 | 70 |
| **4** | Performance | 15 | 0 | 0 | 0 | 15 |
| **5** | GUI | 76 | 24 | 4 | 0 | 104 |
| **6** | Common | 30 | 14 | 18 | 0 | 62 |
| **7** | **Sub total** | 186 | 66 | 30 | 0 | **282** |

➢ **Round 2**

| No | Module code | Pass | Fail | Untested | N/A | Number of test cases |
|---|---|---|---|---|---|---|
| **2** | Use case | 7 | 1 | 0 | 0 | 8 |
| **3** | Use case – Patient | 22 | 1 | 0 | 0 | 23 |
| **4** | Use case – Reminder | 70 | 0 | 0 | 0 | 70 |
| **5** | Performance | 15 | 0 | 0 | 0 | 15 |
| **6** | GUI | 102 | 4 | 0 | 0 | 104 |
| **7** | Common | 62 | 0 | 0 | 0 | 62 |
| **8** | **Sub total** | 276 | 6 | 0 | 0 | **282** |

## 6.7 Test Milestones

| Milestone Task | Resource | Start Date | Finish date |
|---|---|---|---|
| **Test plan** | Test leader | 2015/02/11 | 2015/02/12 |
| **Review Test plan** | Project Manager | 2015/02/13 | 2015/02/13 |
| **Create test case for patient function** | Tester | 2015/02/14 | 2015/02/26 |
| **Create test case for reminder function** | Tester | 2015/02/14 | 2015/02/24 |
| **Create test case for common function** | Tester | 2015/02/27 | 2015/03/02 |
| **Create GUI TC** | Tester | 2015/03/03 | 2015/03/05 |
| **Create performance TC** | Tester | 2015/03/06 | 2015/03/07 |
| **Review test case** | Project Manager | 2015/03/09 | 2015/03/09 |
| **Test case approved** | Project Manager | 2015/03/10 | 2015/03/10 |
| **Test round 1** | Tester | 2015/03/21 | 2015/03/24 |
| **Test round 2** | Tester | 2015/04/03 | 2015/04/03 |
| **Create test report** | Tester | 2015/04/07 | 2015/04/07 |

## 6.8 Defect log

| Defect ID | Module | Description | Type | Severity | Priority | Status | Assign to | Root Cause | Create by | Created Date | Close date | Refer |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 17 | HCP_ Sourcecode | Don't show advice | Coding Logic | Medium | Medium | Closed | AnhDT | wrong SQLite command | AnhDT | 24-Feb-2015 | 26-Feb-2015 | |
| 18 | HCP_ Sourcecode | Don't show google map | Coding Logic | Serious | High | Closed | AnhDT | Don't under stand use Google API | AnhDT | 4-Mar-2015 | 7-Mar-2015 | |
| 19 | HCP_ Sourcecode | Spinner don't display full | Coding Logic | Medium | Medium | Closed | ThienLK | Poor experience design | DuongDT | 24-Feb-2015 | 27-Feb-2015 | |
| 20 | HCP_ Sourcecode | Add new patient fail | Coding Logic | Serious | High | Closed | HuongPT | Lack skill coding | HuongPT | 25-Feb-2015 | 28-Feb-2015 | |
| 21 | HCP_ Sourcecode | Spinner don't display index | Coding Logic | Medium | Medium | Closed | DuongDT | Poor experience design | DuongDT | 24-Feb-2015 | 25-Feb-2015 | |
| 22 | HCP_ Sourcecode | Google map direction wrong | Coding Logic | Serious | High | Closed | AnhDT | Don't under stand use Google API | AnhDT | 4-Mar-2015 | 6-Mar-2015 | |
| 23 | HCP_ Sourcecode | Add new Reminder fail | Coding Logic | Medium | Medium | Closed | HuongPT | Lack skill coding | HuongPT | 4-Mar-2015 | 6-Mar-2015 | |
| 24 | HCP_ Sourcecode | Alarm don't notification | Coding Logic | Serious | High | Closed | HuongPT | Lack skill coding | HuongPT | 7-Mar-2015 | 9-Mar-2015 | |
| 25 | HCP_ Sourcecode | Navigation drawer don't display | Coding Logic | Medium | Medium | Closed | DuongDT | knowleage about fragment | DuongDT | 27-Feb-2015 | 3-Mar-2015 | |
| 26 | HCP_ Sourcecode | Don't show full information of pa | Coding Logic | Medium | Medium | Closed | AnhDT | Command SQLite | AnhDT | 3-May-2015 | 7-Mar-2015 | |
| 27 | HCP_ Sourcecode | Detail reminder can't display | Coding Logic | Medium | Medium | Closed | HuongPT | Command SQLite is problem | HuongPT | 6-Mar-2015 | 8-Mar-2015 | |
| 28 | HCP_ Sourcecode | Maker of hospital don't display | Coding Logic | Serious | Medium | Closed | AnhDT | Lack skill work with google map API | AnhDT | 8-Mar-2015 | 10-Mar-2015 | |
| 29 | HCP_ Sourcecode | Click on maker don't show inforr | Coding Logic | Medium | Medium | Closed | AnhDT | Poor experience design | AnhDT | 12-Mar-2015 | 13-Mar-2015 | |
| 30 | HCP_ Sourcecode | user can't change patient | Coding Logic | Medium | Medium | Closed | HuongPT | Lack skill coding | HuongPT | 10-Mar-2015 | 12-Mar-2015 | |

Figure 6-7 Defect log

# 7 Software User Manual

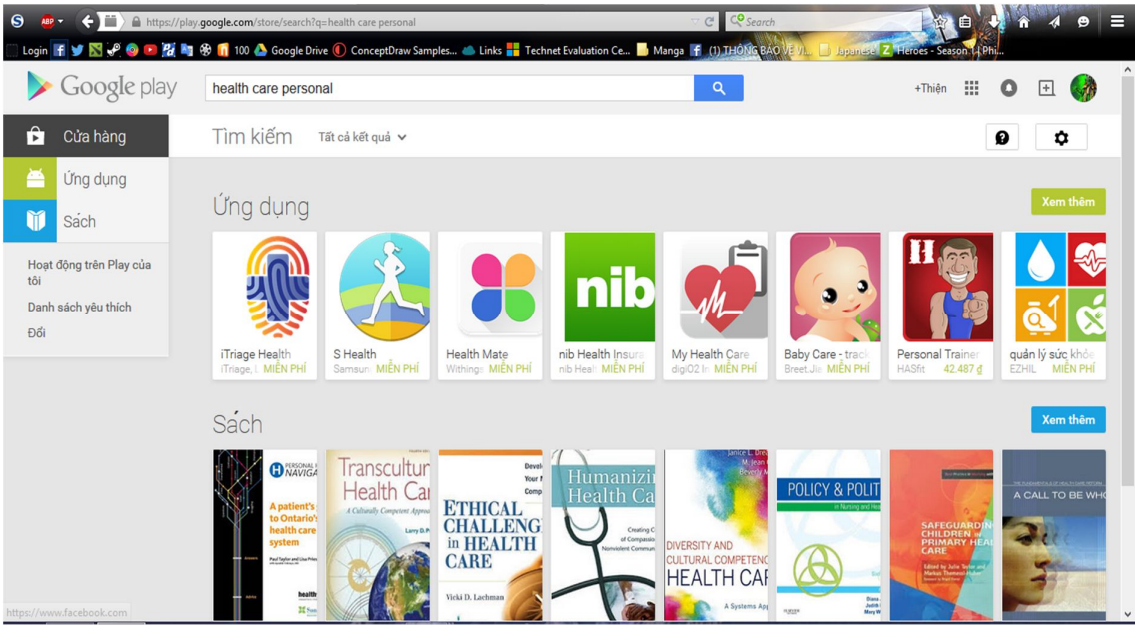
## 7.1 Installation Guide

### 7.1.1 English version

#### 7.1.1.1 Method 1: PC

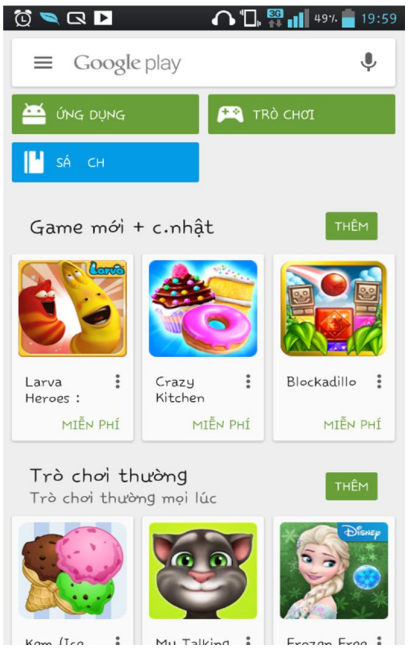
1, Log in to Google Play website https://play.google.com/store by your Gmail account which is used on your device.

2, Search application by name: “Health care personal”




3, Choose application, and click “Install”.

#### 7.1.1.2 Method 2: Directly to the smartphone.




1, Use “CH Play” application.

2, Search application by name: “Health care personal”.

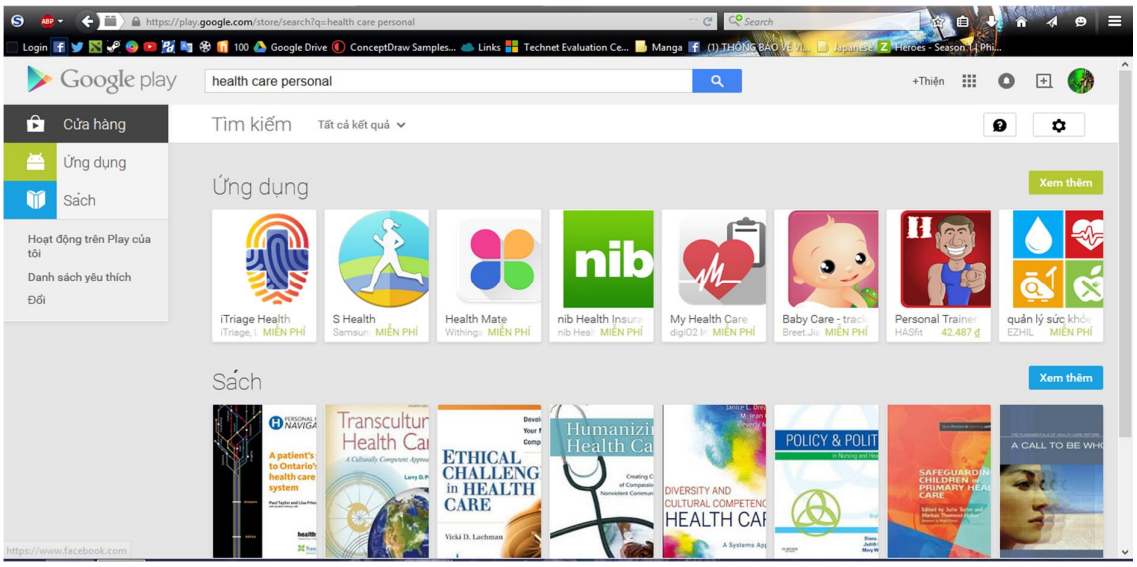
3, Choose application and click “install”

### 7.1.2 Japanese version

#### 7.1.2.1 方法 1: コンピューターとパソコンに関して

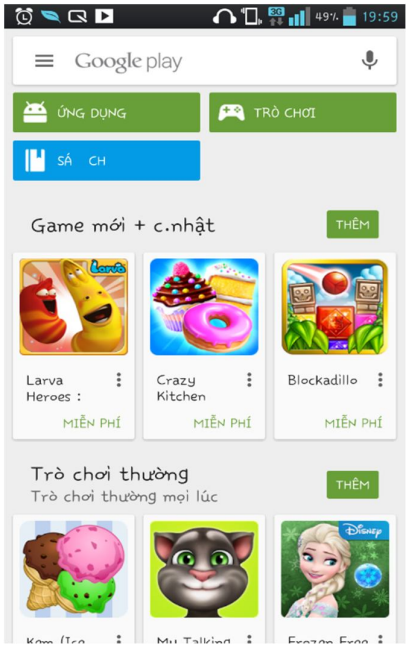
１、お使いのデバイス上で使用されている Gmail アカウントによってウェブサイトの https://play.google.com/store という Google Play にログインします。

２、「Health care personal」：名前でアプリケーションを　検索して。



3、アプリケーションを選択し、「インストール」をクリックします。

#### *7.1.2.2 方法 2: スマート フォンに直接。*




1、 CH Play を使用して。

2、 Heal care personal」：名前でアプリケーションを 検索して。

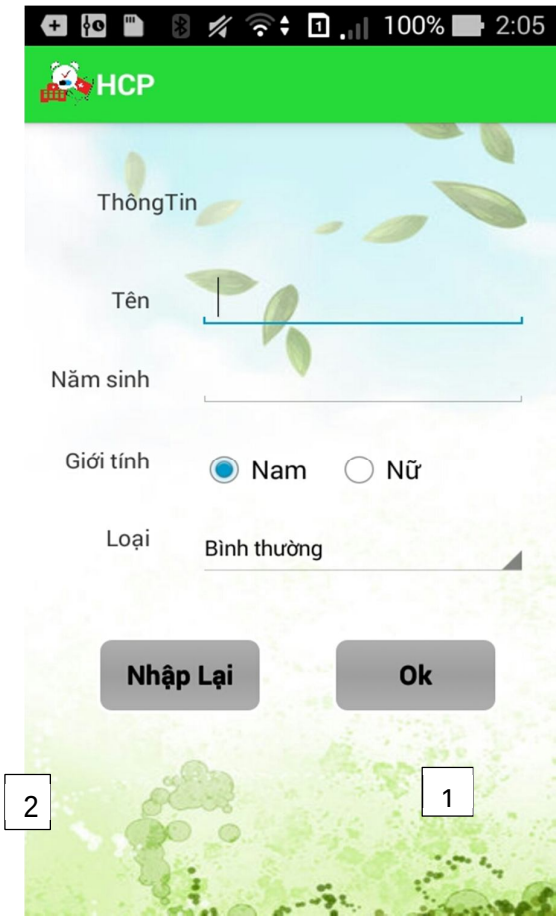
3、 アプリケーションを選択し、「インストール」をクリックします。

## *7.2 User Guide*

### 7.2.1 First time login

- At first time you use the application, you need an account, so application can manage your information, your status and reminder you add. Please add new patient.

### 7.2.2 Add new Patient

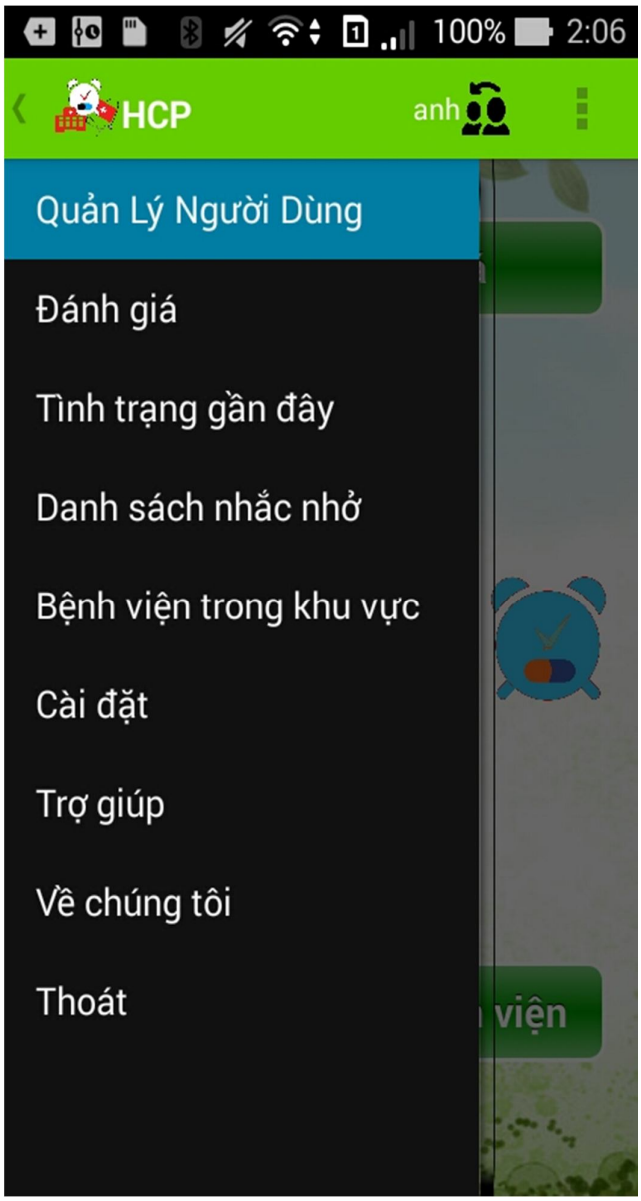



- At the screen, you will see:
  + Information which you need provide to complete to add new patient:
  + “OK”(1) button, click when you complete.
  +”Nhập lại”(2) button, reset information to default.

### 7.2.3 Main screen

- At main screen, you will see
  + Navigation Drawer (1).
  + Current Patient's name (2).
  + "Đánh giá" (3), "Nhắc nhở uống thuốc" (4),"Hiển thị bệnh viện" (5) buttons.
- In the left menu of mobile application, this is Navigation Drawer, which appears in some screen of application. It helps us to enter each field easily and fast. To use Navigation Drawer, we can touch on Navigation Drawer icon or drag out from the left screen.
- Current Patient's name help you know the information of the patient is being used, click to it if you want to switch patient.
- "Đánh giá" (3), "Nhắc nhở uống thuốc"(4),"Hiển thị bệnh viện"(5) buttons, each button represents for 1 function.

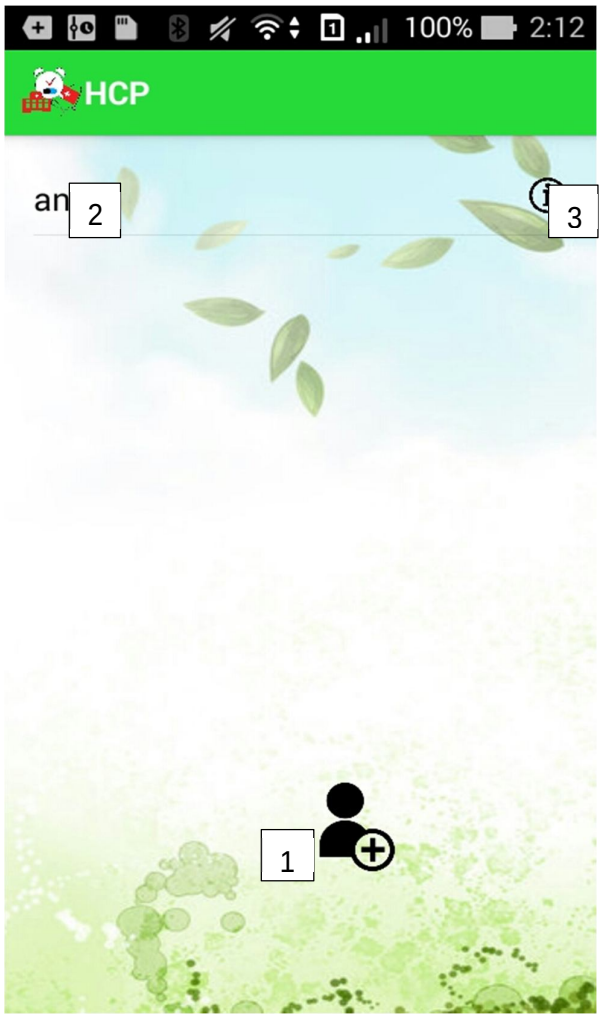
### 7.2.4 Navigation drawer




- At Navigation drawer, you will see some item, each represents for 1 action screen.
- Touch on item and application will display screen.

### 7.2.5 Patient

#### *7.2.5.1 Patient list*

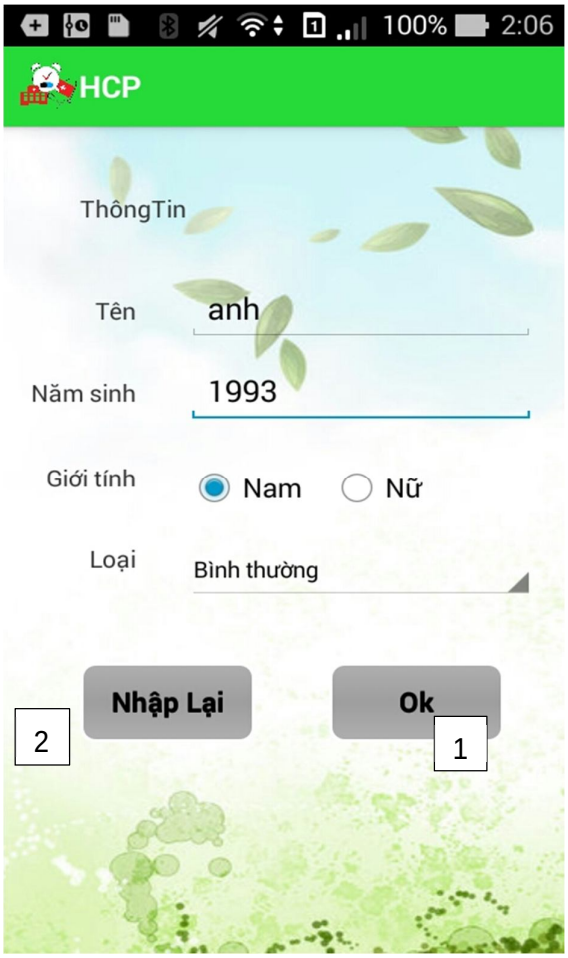



- At Patient list screen, you will see:
  + List patient name with information symbol.
  + Add new button (1).
- You can click on name (2) to switch to that patient, click on information symbol (3) to view detail, and click add button to add new patient.

#### 7.2.5.2 Patient details

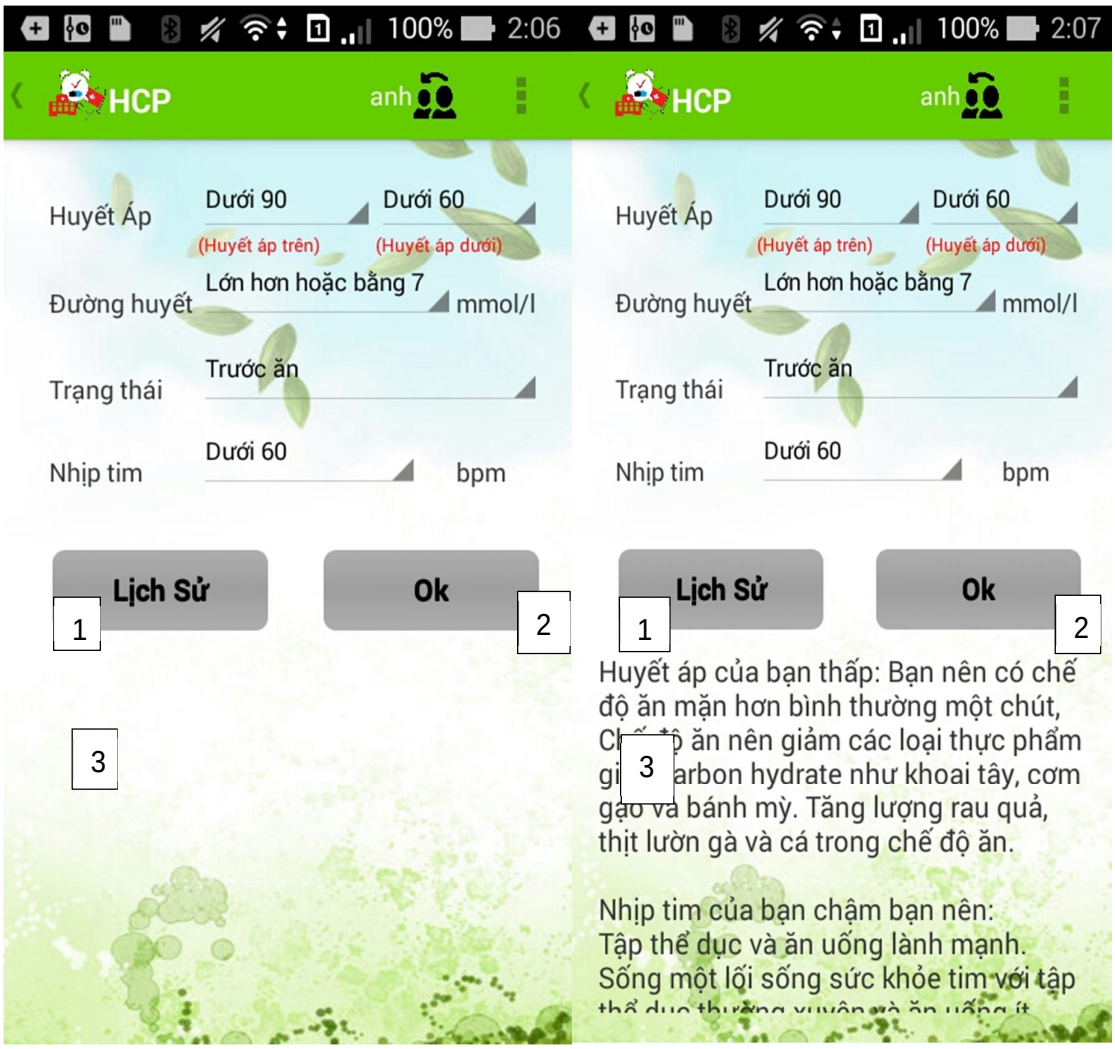
- At patient details screen, you will see:
  + Information of patient.
  + "Xóa"(1) button, click when you want to delete this patient.
  + "Sửa"(2) button, click when you want to edit patient information.
- Patient detail provide basic information about patient.

#### 7.2.5.3 Edit patient information




- Open when you want to edit patient information, at this screen, you will see
  + Already exists Information
  + “OK”(1) button, click when you complete.
  +”Nhập lại”(2) button, reset information to default.

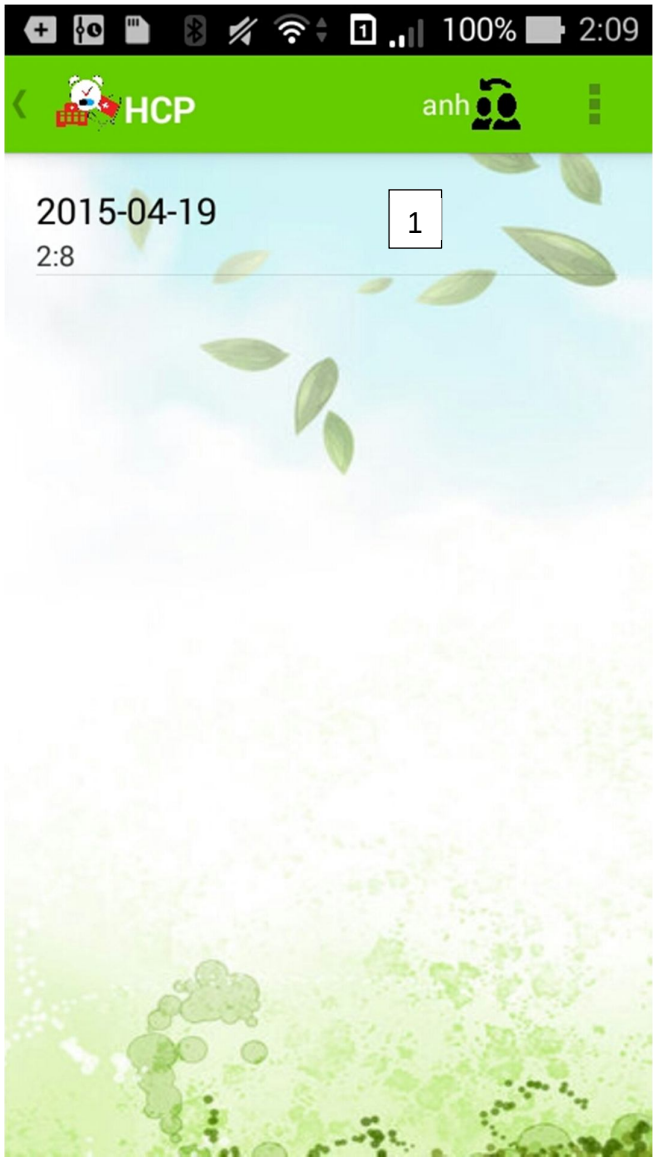
### 7.2.6 Health judgement


HCP
anh
Huyết Áp
Dưới 90
Dưới 60
(Huyết áp trên)
(Huyết áp dưới)
Đường huyết
Lớn hơn hoặc bằng 7
mmol/l
Trạng thái
Trước ăn
Nhịp tim
Dưới 60
bpm
Lịch Sử
Ok
1
2
3
Huyết áp của bạn thấp: Bạn nên có chế độ ăn mặn hơn bình thường một chút,
arbon hydrate như khoai tây, cơm gạo và bánh mỳ. Tăng lượng rau quả, thịt lườn gà và cá trong chế độ ăn.
Nhịp tim của bạn chậm bạn nên:
Tập thể dục và ăn uống lành mạnh.
Sống một lối sống sức khỏe tim với tập

- At Judgment screen you will see:
  + “Huyết áp”, “Đường huyết”, “Trạng thái”, “Nhịp tim” field.
  + “OK”(2) button, click when you complete.
  +”Lịch Sử”(1) button, go to recently status screen.
  + “Đánh giá”(3) box.
- You see some index field and unit of measure, choose them with your index, and application will give advice at “Đánh giá” box after you click “OK” button.

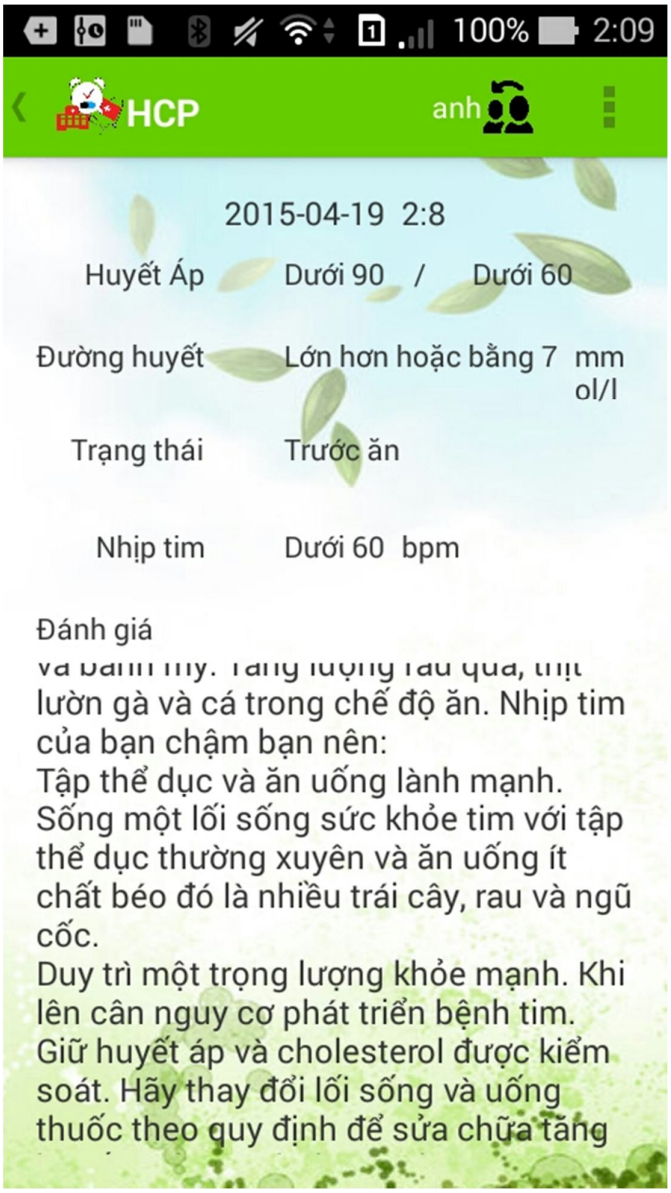
### 7.2.7 Recently status

#### *7.2.7.1 Recently status list*




- After each time patient get a judgment, application will collect it and save to database.
- At Recently status list screen, you can see a list of time, date when a judgment is got by current patient.(1)
- You can click on an item on list to see details.

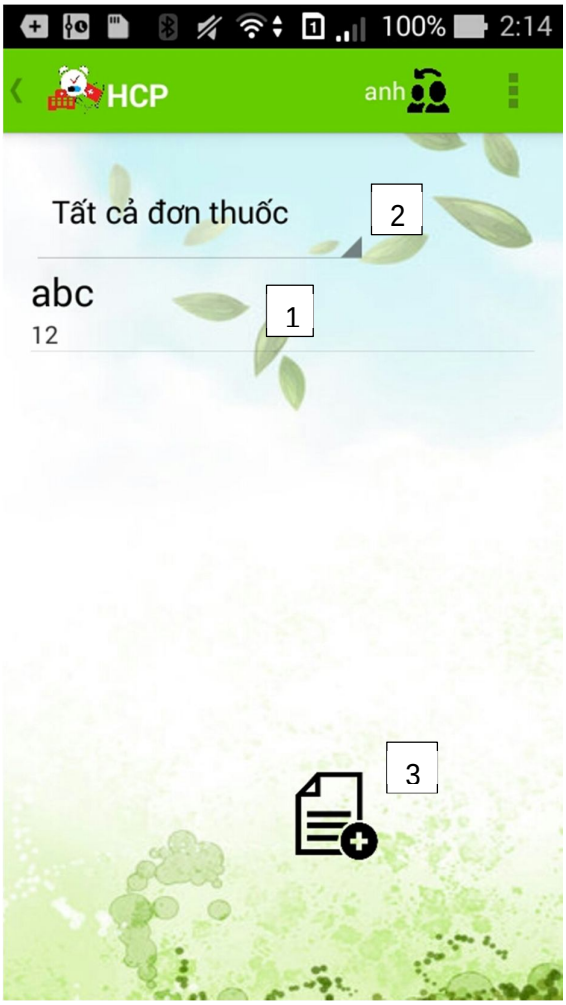
#### 7.2.7.2 Recently status details


100% 2:09
HCP
anh
2015-04-19 2:8
Huyết Áp Dưới 90 / Dưới 60
Đường huyết Lớn hơn hoặc bằng 7 mm ol/l
Trạng thái Trước ăn
Nhịp tim Dưới 60 bpm
Đánh giá
lườn gà và cá trong chế độ ăn. Nhịp tim của bạn chậm bạn nên:
Tập thể dục và ăn uống lành mạnh. Sống một lối sống sức khỏe tim với tập thể dục thường xuyên và ăn uống ít chất béo đó là nhiều trái cây, rau và ngũ cốc.
Duy trì một trọng lượng khỏe mạnh. Khi lên cân nguy cơ phát triển bệnh tim.
Giữ huyết áp và cholesterol được kiểm soát. Hãy thay đổi lối sống và uống thuốc theo quy định để sửa chữa tăng

- At Recently status details screen, you can see information (index and advice) of that session.

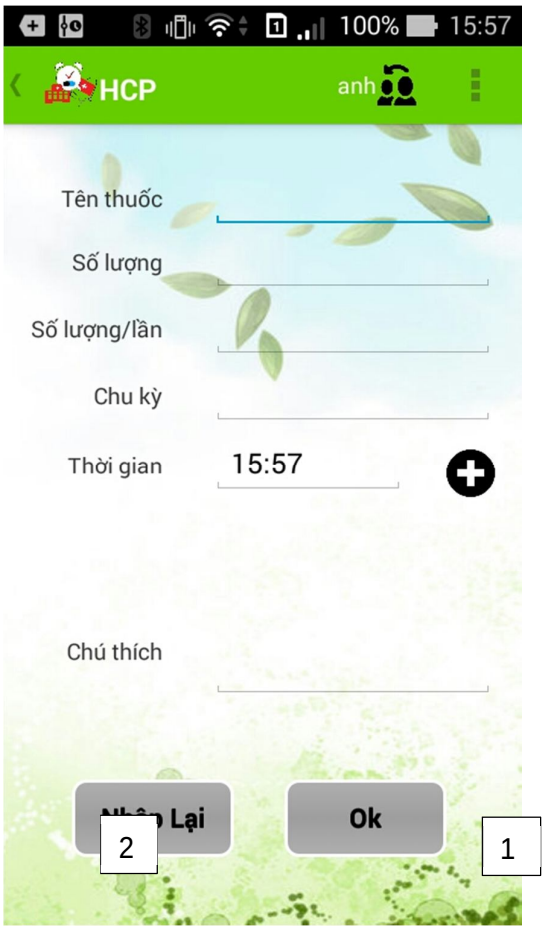
### 7.2.8 Reminder

#### *7.2.8.1 Reminder list*



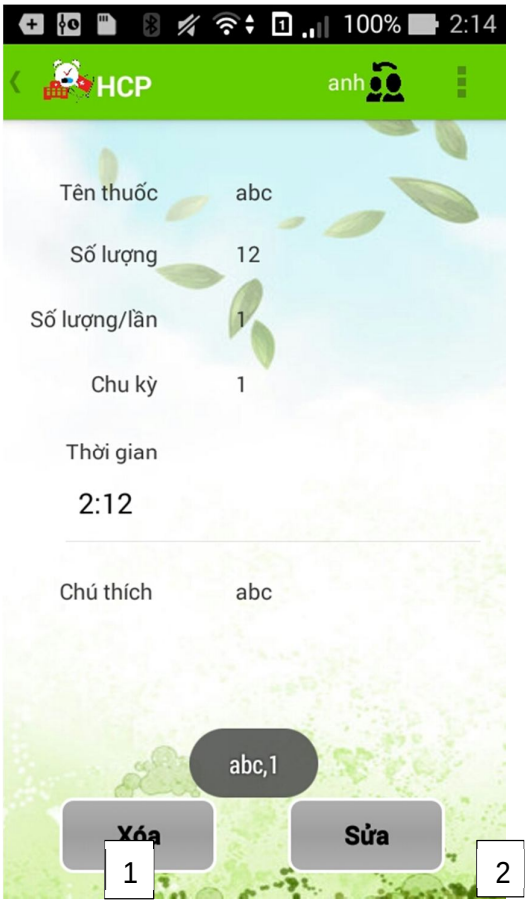

- At Reminder list screen, you will see:
  + List reminder of current Patient.(1)
  + Display type Spinner.(2)
  + Add button.(3)
- You can click on reminder to view detail, or click add button to add new reminder.
- You can choose 2 types to display,
  + "Tất cả đơn thuốc", application displays all of reminder of current patient.
  + "Đơn thuốc hôm nay", application displays reminder which will be alarmed that day of current patient.

#### 7.2.8.2 Add new reminder



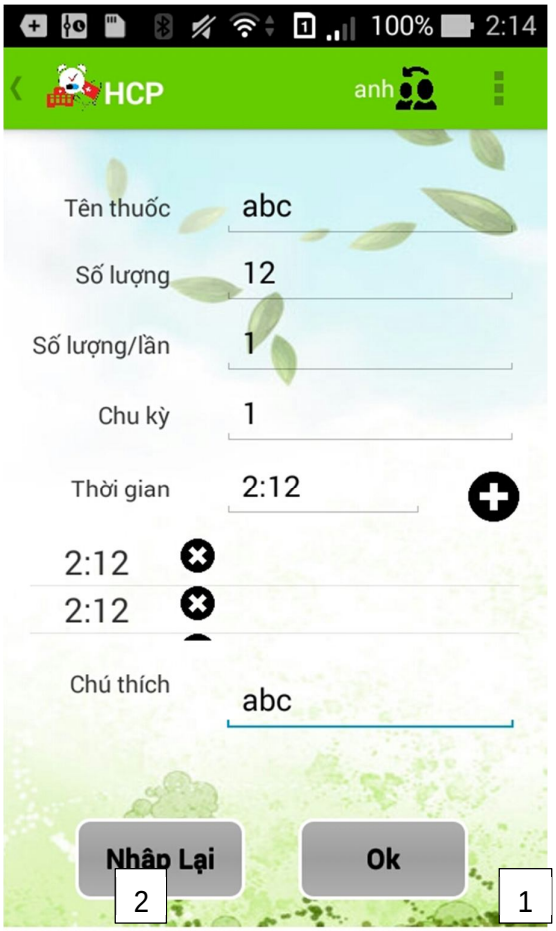

- At Add new reminder screen, you will see:
  + "Tên thuốc", Name of medicine.
  + "Số lượng", Total number medicine you have.
  + "Số lượng/ lần", number medicine each time you get.
  + "Chu kỳ", the number of day between 2 times patient takes the medicine.
  + "Thời gian", time in day you must take this medicine.
  + "Chú thích", some bonus information about the reminder that patient might want to remember beside the information which the function requires.
  + "OK"(1) button, click when you complete.
  +"Nhập lại"(2) button, reset information to default.

#### 7.2.8.3 Reminder details



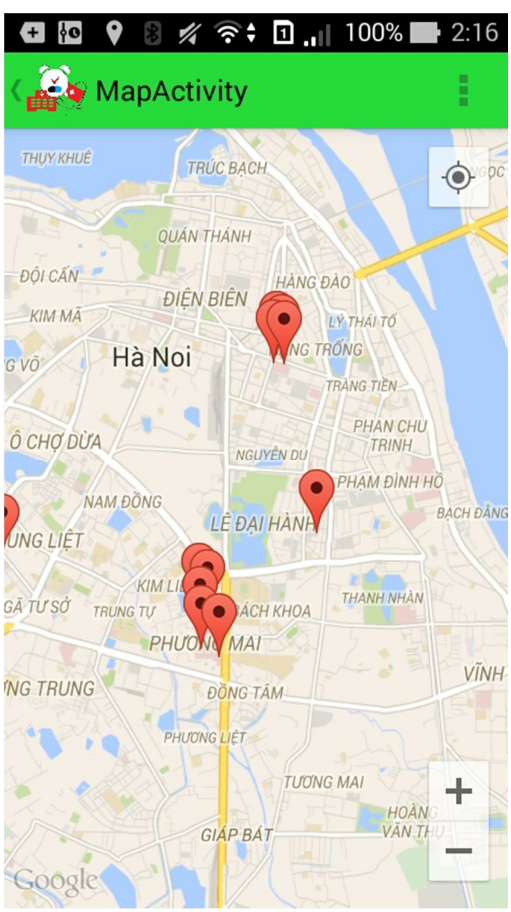

- At Reminder details, you can see information about that reminder.
- + “Xóa”(1) button, click when you want to delete this patient.
- + “Sửa”(2) button, click when you want to edit patient information.

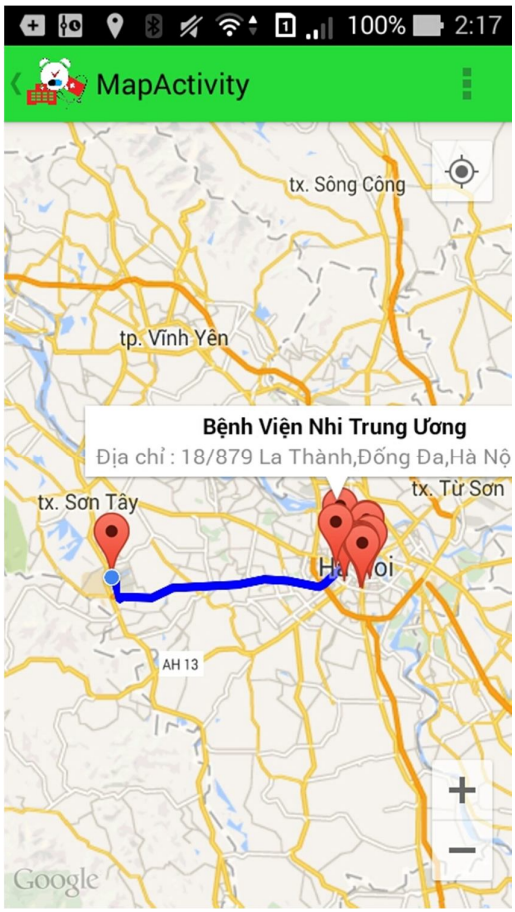
#### 7.2.8.4 Edit reminder




- Open when you want to edit patient information, at this screen, you will see
  + Already information.
  + “OK”(1) button, click when you complete.
  +”Nhập lại”(2) button, reset information to default.

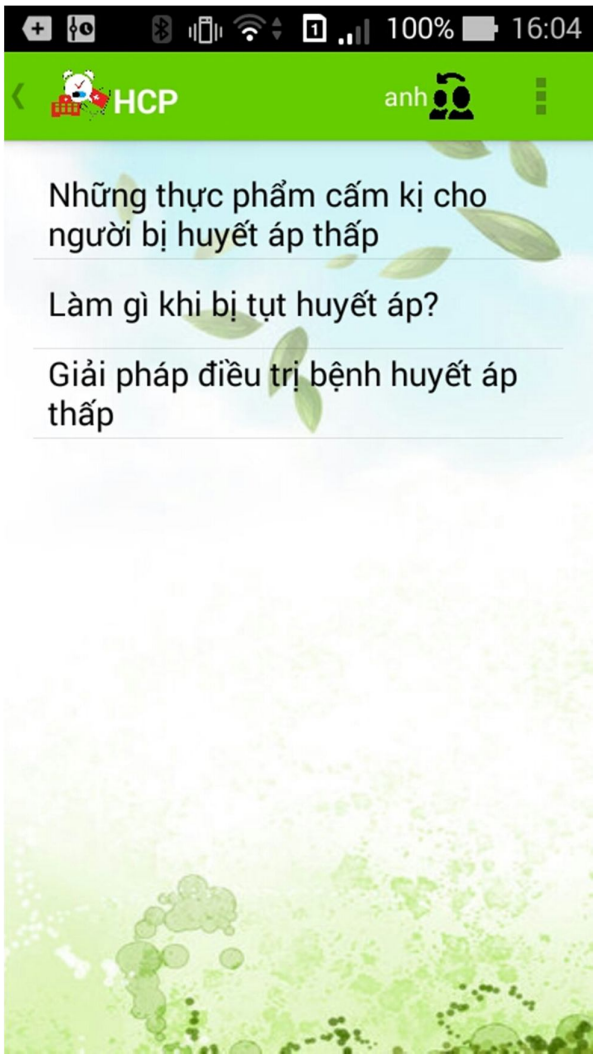
### 7.2.9 Hospital

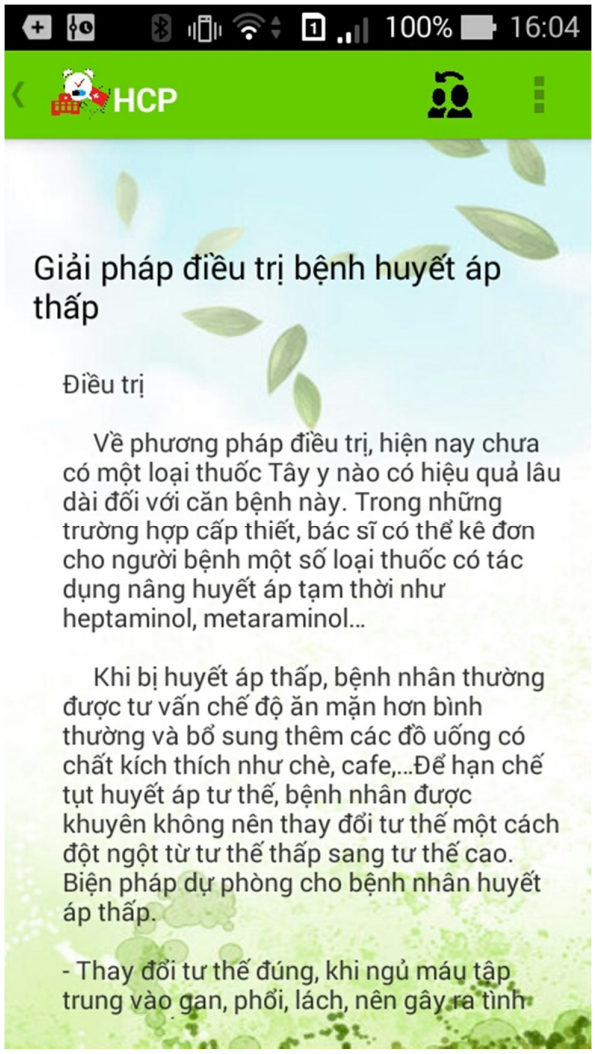



- Function Hospital displays all the hospital in local area.
- User can tab on the marker on the map to show the direction from users location to that hospital (device must turn on GPS/Internet) and also display some basic information of the hospital.

### 7.2.10 Help


100% 16:04
HCP
anh
Những thực phẩm cấm kị cho người bị huyết áp thấp
Làm gì khi bị tụt huyết áp?
Giải pháp điều trị bệnh huyết áp thấp


100% 16:04
HCP
Giải pháp điều trị bệnh huyết áp thấp
Điều trị
Về phương pháp điều trị, hiện nay chưa có một loại thuốc Tây y nào có hiệu quả lâu dài đối với căn bệnh này. Trong những trường hợp cấp thiết, bác sĩ có thể kê đơn cho người bệnh một số loại thuốc có tác dụng nâng huyết áp tạm thời như heptaminol, metaraminol...
Khi bị huyết áp thấp, bệnh nhân thường được tư vấn chế độ ăn mặn hơn bình thường và bổ sung thêm các đồ uống có chất kích thích như chè, cafe,...Để hạn chế tụt huyết áp tư thế, bệnh nhân được khuyên không nên thay đổi tư thế một cách đột ngột từ tư thế thấp sang tư thế cao.
Biện pháp dự phòng cho bệnh nhân huyết áp thấp.
- Thay đổi tư thế đúng, khi ngủ máu tập trung vào gan, phổi, lách, nên gây ra tình

- Function Advisory provides posts which contents useful information about heath care or some diseases. The posts will be updated each time the application be updated.

# 8 Appendixes

## 8.1 Meeting minutes

**WEEKLY MEETING MINUTES**

**Project Name:** Health Care Personal
**Project Code:** *HCP*
**Project Manager:** *Dang Tuan Anh*
**Conductor:** *Pham Ngoc Ha*
**Secretary:** *Nguyen Cam Linh*
**Date:** *27/01/2015*
**Venue:**

**List of Attendees**

| No | Full Name | Unit/Group | Position | Attendance |
|---|---|---|---|---|
| 1 | Pham Ngoc Ha | FU | Instructor | Present |
| 2 | Dang Tuan Anh | FU | PM | Present |
| 3 | Duong Thai Duong | FU | PTL | Present |
| 4 | Pham Thien Huong | FU | Dev | Present |
| 5 | Le Khanh Thien | FU | Tester | Present |
| 6 | Nguyen Cam Linh | FU | Tester | Present |

**Content**

| No | Agenda | Detail & Key points | PIC | Deadline |
|---|---|---|---|---|
| 1 | Main Tasks performed this week | *- Create test case for common function* | ThienLK, LinhNC | 2/3/2015 |
| | | *- Create GUI TC* | ThienLK, LinhNC | 5/3/2015 |
| | | *- Develop display hospital* | AnhDT | 12/3/2015 |
| | | *- Create performance TC* | ThienLK | 7/3/2015 |
| | | *- Review test case* | AnhDT | 9/3/2015 |

Figure 8-1 Meeting minutes English version

**WEEKLY MEETING MINUTES**

**Project Name:** Health Care Personal
**Project Code:** *HCP*
**Project Manager:** *Dang Tuan Anh*
**Conductor:** *Pham Ngoc Ha*
**Secretary:** *Nguyen Cam Linh*
**Date:** *27/01/2015*
**Venue:**

**List of Attendees**

| No | Full Name | Unit/Group | Position | Attendance |
|---|---|---|---|---|
| 1 | Pham Ngoc Ha | FU | Instructor | Present |
| 2 | Dang Tuan Anh | FU | PM | Present |
| 3 | Duong Thai Duong | FU | PTL | Present |
| 4 | Pham Thien Huong | FU | Dev | Present |
| 5 | Le Khanh Thien | FU | Tester | Present |
| 6 | Nguyen Cam Linh | FU | Tester | Present |

**Content**

| No | Agenda | Detail & Key points | PIC | Deadline |
|---|---|---|---|---|
| 1 | Main Tasks performed this week | *一般的なファンクションに関するテストケースを設定する* | ThienLK, LinhNC | 2/3/2015 |
| | | *- Create GUI TC* | ThienLK, LinhNC | 5/3/2015 |
| | | *[hospital]に関する部分を画面に表示する開発* | AnhDT | 12/3/2015 |

Figure 8-2 Meeting minutes Japanese version

## 8.2 Progress report

Fpt University

**PROJECT PROGRESS REPORT1**

| | | | |
|---|---|---|---|
| **Project Name** | Health care personal | **Project Code** | HCP |
| **Author** | Dang Tuan Anh | **Project Manager** | Dang Tuan Anh |
| **Date of Report** | 19/01/2015 | **Receiver** | Lecturer: Pham Ngoc Ha |
| **Reporting period** | 2 weeks | | |

**Progress Description**

| Items | Information | Note |
|---|---|---|
| **Start-date of project** | 05/01/2015 | |
| **Estimated end-date** | 15/4/2015 | |
| **Team size** | 5 members | Le Khanh Thien – SE02447<br>Dang Tuan Anh – SE02643<br>Pham Thien Huong –SE02880<br>Duong Thai Duong – SE02570<br>Nguyen Cam Linh –SE02565 |
| **Total estimated effort** | 1728 man – hours | |
| **Total effort spent** | 120 man – hours | |
| **Effort spent in this period** | 120 man – hours | |
| **Total effort left** | 1608 man – hours | |

**Customer Complaints: None**

**Tasks matches/missed**

| Task | | Responsibility | Deliverable | Status | Remark | Re-schedule |
|---|---|---|---|---|---|---|
| Selecting and unifying the project's topic | | All members | Selected topic | Done | Determine and unify topic | |
| Reviewing the software development process | | All members | Record | Done | Review all works in the process | |
| Training Android programing | | AnhDT, DuongDT, HuongPT | Sample program Android | In progress | | Continue on next week |
| Studying SQLite database for Android | | AnhDT, DuongDT, HuongPT | Sample that implemented SQLite | In progress | Study the basic usage of SQLite in Android | Continue on next week |
| Report 1 | Writing Software requirements specification document | | SRS document | Done | | |
| | Writing project plan | | Project plan document, WBS | Done | | |
| | Writing Q&A management (version JP & E) | | Q&A management document (version JP & E) | Done | | |

**Tasks planned for next period**

| Task | Deliverable | Planned end date |
|---|---|---|
| Create Architecture Design | Architecture Design | 02/02/2015 |

Figure 8-3 Progress report English version

Fpt University

# プロジェクト進捗報告書１

| | | | |
|---|---|---|---|
| プロジェクト名 | Health care personal | プロジェクトコード | HCP |
| 作成者 | Dang Tuan Anh | プロジェクトリーダ | Dang Tuan Anh |
| 日付 | 2015年 1月 19日 | 受領者 | Lecturer: Pham Ngoc Ha |
| 報告期間 | 二週間 | | |

## 進捗説明

| アイテム | 情報 | 注意 |
|---|---|---|
| プロジェクトの開始日 | 05/01/2015 | |
| 推定終了日 | 15/4/2015 | |
| チームのサイズ | 5人 | Le Khanh Thien – SE02447<br>Dang Tuan Anh – SE02643<br>Pham Thien Huong – SE2880<br>Duong Thai Duong – SE02570<br>Nguyen Cam Linh – SE02555 |
| 推定総努力 | 1728 工数 | |
| 消費された総努力 | 120 工数 | |
| この期間で消費された努力 | 120 工数 | |
| 残り総努力 | 1608 工数 | |

## 顧客等からの苦情: なし

## Tasks matches/missed

| タスク | | 責任 | 成果物 | 状況 | Remark | リスケジュール |
|---|---|---|---|---|---|---|
| 選択し、プロジェクトのトピックを統一する | | 全員 | 選択したトピック | 完成 | 決定し、トピックを統一する | |
| ソフトウェア開発工程をレビューする | | 全員 | 記録 | 完成 | すべての作業をレビューする | |
| Androidプログラミング言語を養成する | | AnhDT, DuongDT, HuongPT | サンプルの Android アプリ | 進行中 | | 来週に進む |
| Android用の SQLite データベースを研究する | | AnhDT, DuongDT, HuongPT | Sample that implemented SQLite | 進行中 | SQLite の基本的な使い方を学ぶ | 来週に進む |
| Report 1 | 要求仕様書を書く | | 要求仕様書 | 完成 | | |
| | プロジェクト計画を書く | | プロジェクト計画, WBS | 完成 | | |
| | Q&A 管理ドキュメントを書く（日本語・英語版） | | Q&A 管理ドキュメント（日本語・英語版 | 完成 | | |

## 次の期間のタスク

| タスク | 成果物 | 推定終了日 |
|---|---|---|
| アーキテクチャ設計を作成する | アーキテクチャ設計 | 02/02/2015 |
| 画面のデザインを作成する | 画面デザイン | 02/02/2015 |

Figure 8-4 Progress report Japanese version

## 8.3 Q&A

Q&A管理シート

| | |
|---|---|
| プロジェクトコード: | HCP |
| Q&A管理者: | Le Khanh Thien |
| 最終更新日 | 6/Jan/15 |

| | |
|---|---|
| オープン項目数 | 0 |
| 確認中項目数 | 0 |
| 確認済項目数 | 16 |
| キャンセル項目数 | 0 |
| 合計 | 16 |

| 項番 | ロケーション | 質問 | 回答 | 重要度 | 作成日 | 作成者 | 回答者 | 回答期限 | 回答日 | ステータス | 質問種別 | プロセス |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | どのようにコードの行を推定するには？ | 最初の実数やメソッドの数を見積もる。その後、方法を分析. | High | 6/Jan/15 | ThienLK | HaPN | 13/Jan/15 | 10/Jan/15 | Closed | Clarification | Requirement |
| 2 | Project plan | In the future, can application intergrate with some device which can measure user's index. | May be, that's future's problem. For now, that's out of developer's ability | Low | 6/Jan/15 | ThienLK | HaPN | 10/Jan/15 | 7/Jan/15 | Closed | Customer Defect | Requirement |
| 3 | SRS | How to get the location of the hospital? | The location of the hospital is contained in the database with the index of longitude and latitude. And when the Map function want the display a hosplital as a marker on the map, it search by the longitude and latitude of the hospital | High | 6/Jan/15 | HuongPT | AnhDT | 13/Jan/15 | 10/Jan/15 | Closed | Clarification | Design |
| 4 | SRS | Can user delete a Recently status? Why not? | User can't because the developer design it as a history, so it should be remain for user to keep trace of their health's index by time | High | 7/Jan/15 | AnhDT | HuongPT | 10/Jan/15 | 7/Jan/15 | Closed | Confirmation | Requirement |
| 5 | Test case | How to test alarm of reminder function? | Set the time in the Reminder, then wait until that time comes | Medium | 14/Feb/15 | LinhNC | HuongPT | 10/Mar/15 | 1/Mar/15 | Closed | Clarification | Test |
| 6 | Test case | When display the List of Reminder, the application should display all the reminder or just the reminders which belongs to the current patient or by any specific way? | The function should just display the reminders which belong to the current function. There should be more other option for user to choose to display the Reminder's List than just display all such as display the reminder which will be alarmed today (The day user is using the application) | High | 1/Feb/15 | HuongPT | HaPN | 9/Feb/15 | 3/Feb/15 | Closed | Clarification | Requirement |

Figure 8-5 Q&A English version

Q&A管理シート

| | |
|---|---|
| プロジェクトコード: | HCP |
| Q&A管理者: | Le Khanh Thien |
| 最終更新日 | 6/Jan/15 |

| | |
|---|---|
| オープン項目数 | 0 |
| 確認中項目数 | 0 |
| 確認済項目数 | 11 |
| キャンセル項目数 | 0 |
| 合計 | 11 |

| 項番 | ロケーション | 質問 | 回答 | 重要度 | 作成日 | 作成者 | 回答者 | 回答期限 | 回答日 | ステータス | 質問種別 | プロセス |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | どのようにコードの行を推定するには？ | 最初の実数やメソッドの数を見積もる。その後、方法を分析 | 高 | 1/6/2015 | ThienLK | HaPN | 1/13/2015 | 1/10/2015 | 確認済 | 明確化 | 要求 |
| 2 | Project plan | 将来的には、アプリケーションは、ユーザーのインデックスを測定することができ、いくつかのデバイスと統合することができますか。 | それが将来の問題です。今のところ、それは開発者の能力の外だ | 低 | 1/6/2015 | ThienLK | HaPN | 1/10/2015 | 1/7/2015 | 確認済 | 顧客側不具合 | 要求 |
| 3 | SRS | どのように病院の場所を取得するには？ | 病院の位置は、緯度と経度のインデックスを使用してデータベースに含まれている。<br><br>マップ機能は、ディスプレイにマップ上のマーカーとして病院にしたいとき、それは病院の経度と緯度で検索. | 高 | 1/6/2015 | HuongPT | HaPN | 1/13/2015 | 1/10/2015 | 確認済 | 明確化 | 設計 |
| 4 | SRS | ユーザーは、最近の状況を削除できますか？ | 開発者が履歴として設計するので、ユーザはできない。<br>従って、時間によってそれらの健康指標を追跡するために、ユーザに残るであるべきである | 高 | 1/7/2016 | AnhDT | HaPN | 1/10/2015 | 1/7/2015 | 確認済 | 確認 | 要求 |

Figure 8-6 Q&A Japanese version

# 9 References

[1]. Iterative model:

http://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm

http://en.wikipedia.org/wiki/Iterative_and_incremental_development

[2]. Sparx system, model:

http://www.sparxsystems.com/resources/tutorial/logical_model.html

http://www.sparxsystems.com/resources/uml2_tutorial/uml2_usecasediagram.html

http://www.sparxsystems.com/resources/tutorial/use_case_model.html

http://www.sparxsystems.com/resources/tutorial/dynamic_model.html

[3]. Wikipedia, 'Use case'

http://en.wikipedia.org/wiki/Use_case

[4]. MSDN, ' UML Class Diagrams: Guidelines

http://msdn.microsoft.com/en-us/library/dd409416.aspx

[5]. MSDN, ' UML Sequence Diagrams: Guidelines ',

http://msdn.microsoft.com/en-us/library/dd409389.aspx

[6]. Fer.unizg, Android coding convention:

https://www.fer.unizg.hr/_download/repository/ProjectConventions_-_Java_Android.pdf

[7]. Nguyen Trung Dung, RTF_Report_3_SRS, (2014/01), FPT University, Hanoi, Vietnam.

[8]. VHP group, VHP_FinalReport, (2014/08/17), FPT University, Hanoi, Vietnam.

[9]. EC group, EC_FinalReport, (2014/12/12), FPT University, Hanoi, Vietnam.